CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE

Expression et analyse des besoins en UML avec PowerAMC

Manuel d'utilisation

Version 1.1

Février 2005

Réalisé par
la Direction des systèmes d'information du CNRS (DSI)

Table des matières

TABLE DES MATIERES _____ **3**

A PROPOS DE CE MANUEL _____ **5**

MISES A JOUR _____ **7**

PRESENTATION GLOBALE DE LA METHODE _____ **9**

VUE METIER _____ **11**

1. **Les acteurs** **11**
2. **Les processus métier** **13**
3. **Le modèle du domaine** **16**

VUE SYSTEME INFORMATIQUE _____ **17**

1. **Le contexte statique** **17**
2. **Les cas d'utilisation** **19**
3. **Le contexte dynamique** **30**
4. **Le modèle du domaine affiné** **32**

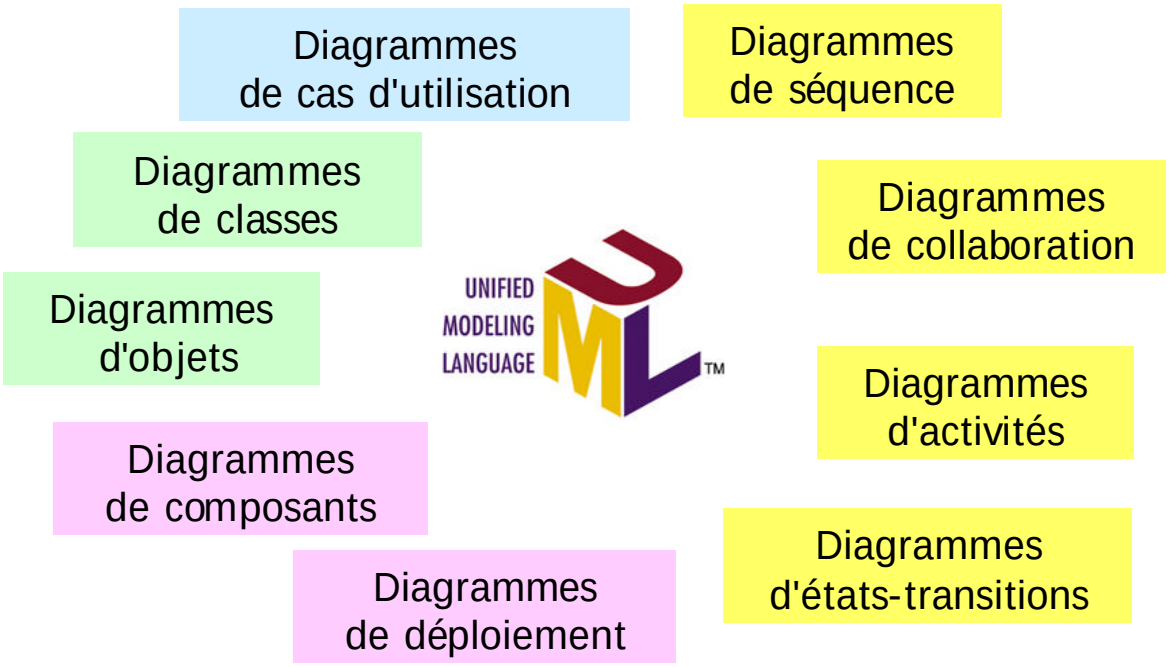
VUE APPLICATIVE _____ **37**

1. **La maquette** **37**
2. **La navigation (optionnel)** **39**
3. **Les classes participantes (optionnel - début de conception de l'architecture)** **41**

A propos de ce manuel

Ce manuel d'utilisation a pour but de décrire la méthode d'expression et d'analyse des besoins, utilisant le langage de modélisation **UML** et supportée par l'outil **PowerAMC**. Il s'adresse aux concepteurs des équipes projet, qui partent d'une expression initiale des besoins par des utilisateurs ou des maîtres d'ouvrage.

UML (Unified Modeling Langage) n'est pas une méthode de développement mais un **langage de modélisation** qui définit des standards relatifs à la modélisation orientée objet. Ce langage permet de mettre en œuvre neuf diagrammes différents. Dans la mise en œuvre d'une méthode de développement, l'un ou l'autre des diagrammes est choisi en fonction des concepts que l'on veut représenter à une étape de développement donnée. Les concepts permettent de couvrir les étapes, depuis l'expression des besoins jusqu'au codage.



La démarche définie dans le présent guide, a pour objectifs de couvrir l'expression et l'analyse des besoins ; elle utilise **cinq diagrammes UML** : deux diagrammes statiques (diagrammes de classes et de cas d'utilisation), trois diagrammes dynamiques (diagramme d'activités, de collaboration et de séquence).

L'expression et l'analyse des besoins se matérialise par des documents Word de différents niveaux, dans lesquels on insère des diagrammes UML produits à l'aide de l'outil PowerAMC. Les plans types de ces documents sont disponibles depuis le site développement Web : http://www.dsi.cnrs.fr/bureau_qualite/developpement-web/guides-modeles/guides-modeles.asp :

- Note de cadrage (inclut des diagrammes de la vue métier)
- Exigences fonctionnelles (inclut des diagrammes de la vue système informatique)
- Conception de l'interface utilisateur (inclut des diagrammes de la vue applicative)

Remarques :

- Le terme **stéréotype** est fréquemment utilisé dans ce guide et mérite une définition préalable : « Extension du vocabulaire UML, qui permet de créer de nouvelles variétés d'éléments constitutifs dérivés d'éléments existants, mais qui sont spécifiques à votre problème. » [PowerAMC MOO – Guide de l'utilisateur].
  Exemple d'utilisation de stéréotypes : dans la première étape de la démarche, qui s'intéresse aux processus métier du domaine étudié, le concept «processus métier » n'étant pas directement supporté par UML, un stéréotype de cas d'utilisation est utilisé.
- Le terme **système** est utilisé dans ce document pour désigner de manière globale l'application informatique.

Utilisation de ce manuel

- Le premier chapitre « Présentation globale de la méthode » donne un aperçu global de la méthode utilisée et montre la progressivité de l'approche.
- Les autres chapitres se lisent a priori dans l'ordre, chacun d'entre eux décrit une étape de la démarche de modélisation. Naturellement, le concepteur peut être amené à revenir sur l'un ou l'autre des chapitres dans le cadre des itérations successives qu'il va mettre en œuvre pour son projet. Chaque chapitre sépare la présentation des concepts manipulés et leur représentation avec l'outil PowerAMC.

Utilisation de l'outil PowerAMC

- Un **modèle type** PowerAMC est mis à la disposition des concepteurs. Ce modèle permet de démarrer une modélisation avec les packages et les diagrammes déjà créés, ainsi que quelques exemples de concepts dans chaque diagramme. Le nom du modèle est : *Modèle – application web.moo*. Ce modèle contient également un exemple de rapport, produit à l'aide d'un modèle de rapport (il inclut les diagrammes créés dans chacune des vues, ainsi que les classes définies avec leur liste d'attributs et d'opérations).
- Par ailleurs, des **exemples de projets** modélisés selon la démarche contenue dans le présent guide sont disponibles. Les documents Word et modèles PowerAMC sont mis à disposition des équipes DSI dans le répertoire *PROJETS de TOULOUSE Uml/DSI/Démarche UML DSI.*
- Pour accéder au modèle type et aux exemples PowerAMC, le fichier à ouvrir est l'espace de travail : « **Espace de travail - Démarche UML DSI.sws** ».

Conventions

Les noms de fenêtre, de zone de dialogue ou de zone de saisie apparaissent entre guillemets. Les noms de commande d'un menu apparaissent en gras italique séparés par des /, par exemple : menu ***Fichier/Imprimer…***

Ce pictogramme identifie des remarques utiles mais sans incidence sur le cours des instructions exposées.

Ce pictogramme identifie des informations à lire et/ou à exécuter **impérativement** qui peuvent influencer le cours des instructions suivantes.

Assistance

Pour des compléments d'information, vous pouvez contacter le groupe de travail UML à la DSI : « Liste Uml(at)dsi.cnrs.fr ».

Mises à jour

V 0.1	**Septembre 2003**	Version initiale du manuel d'utilisation PowerAMC pour l'expression des besoins avec UML, créée suite à la formation du groupe de travail.
V1.0	**Janvier 2005**	Création d'un manuel séparé de présentation de PowerAMC. Mise à jour de la démarche suite aux retours d'expérience de projets pilotes et ajouts de compléments issus de l'état de l'art (en particulier, conseils pour la rédaction des cas d'utilisation).
V1.1	**Février 2005**	Ajout d'un chapitre «Modèle du domaine affiné » dans la vue système informatique, décrivant les concepts du diagramme de classe et mise à niveau du chapitre « Modèle du domaine ».

Pour une première prise en main de PowerAMC, pour quelques trucs et astuces, pour en savoir plus sur les rapports, ou pour connaître la liste des stéréotypes définis, consulter le « Manuel d'utilisation de PowerAMC : Généralités-Rapports-Stéréotypes », accessible dans le répertoire *PROJETS de TOULOUSE Uml/DSI/PowerAMC*

ou depuis le site développement Web :

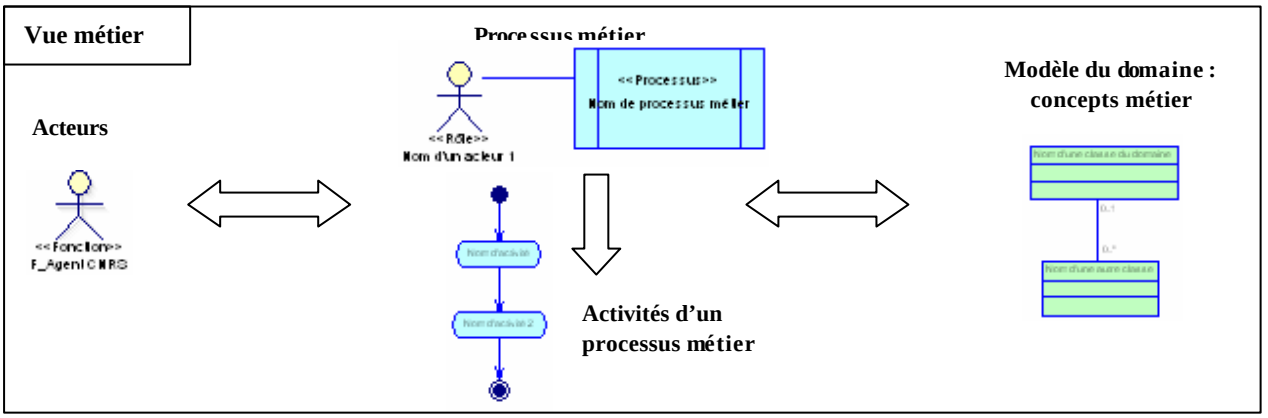
http://www.dsi.cnrs.fr/bureau_qualite/developpement-web/guides-modeles/guides-modeles.asp.

Avec la version 9 de PowerAMC, **ne pas utiliser de fonctions sur les symboles** car ces fonctions provoquent des dégradations du modèle irréversibles.

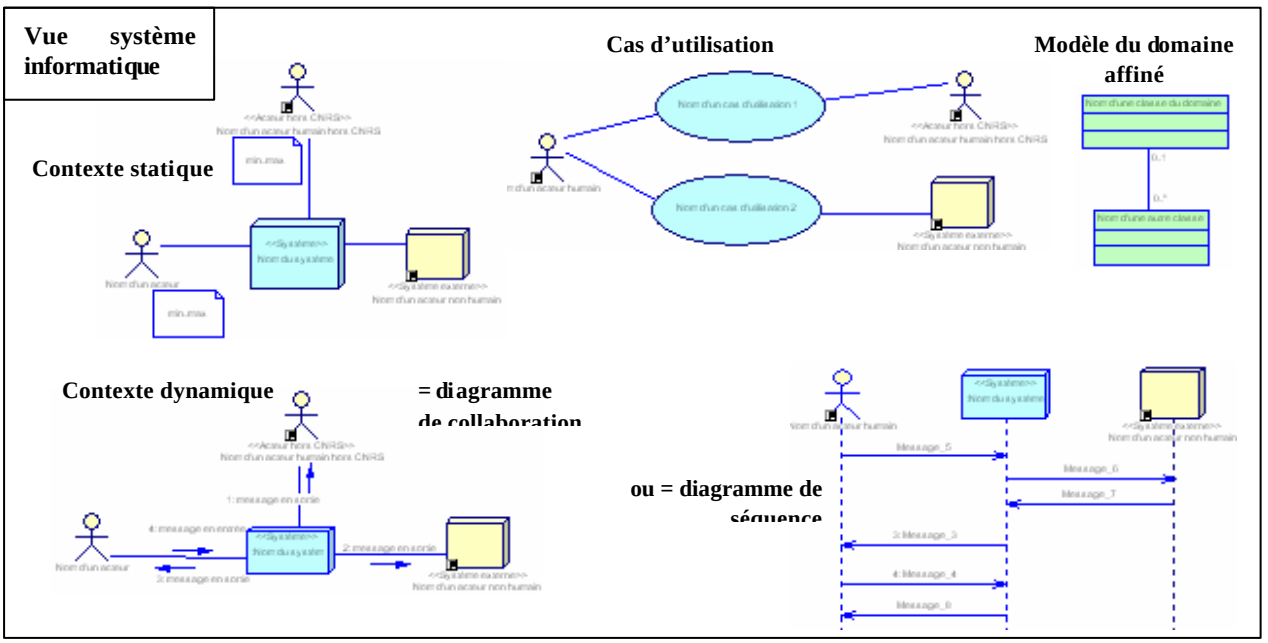
Ces fonctions sont accessibles à partir du menu principal ***Symbole/*** ou bien à partir du menu contextuel sur un objet (clic droit de la souris et ***Disposition*** ou ***Cacher le symbole***…).

Présentation globale de la méthode

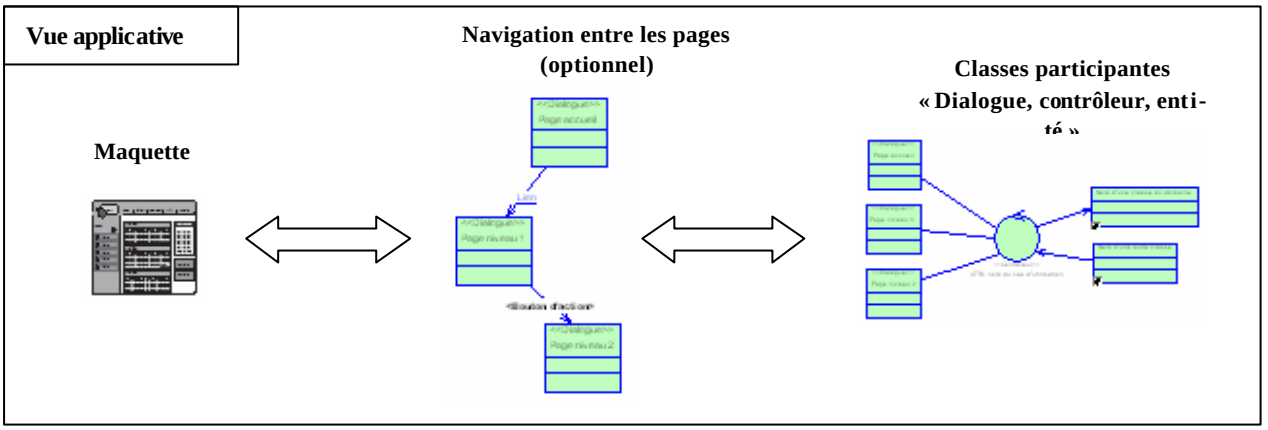
La première étape de la démarche consiste à mieux connaître et comprendre les processus dans lesquels va s'intégrer le futur système informatique. C'est ce qu'on appelle la **vue métier**.
Il s'agit à ce niveau d'identifier les acteurs, les processus et les concepts métiers qui composent le domaine étudié. Les concepts métier sont modélisés sous forme de classes. La description des processus métier permet de définir précisément le périmètre que couvrira le futur système informatique, c'est-à-dire les activités qui vont faire l'objet d'une automatisation.



L'étape suivante s'intéresse au système informatique. On représente dans un premier temps le contexte d'utilisation : le système vu comme une boîte noire et les acteurs qui interagissent avec lui (contexte statique). Ensuite on décrit les services rendus aux différents utilisateurs sous forme de cas d'utilisation. Il est possible de compléter ces descriptions par des représentations de la dynamique des échanges de messages entre le système et les utilisateurs (contexte dynamique). Ces différentes représentations forment la **vue système informatique**.



Enfin, la **vue applicative** permet de donner une vision concrète du futur système grâce à une maquette et à la modélisation des principes de navigation entre les pages. Egalement, il est possible de représenter un début d'architecture applicative : la séparation entre les couches Dialogue, Traitements applicatifs et Données.

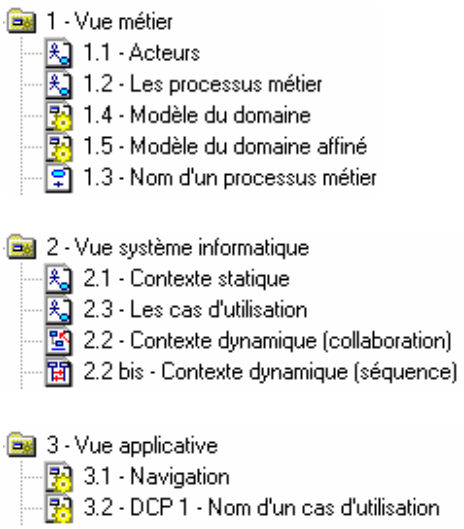


En pratique, les diagrammes de la vue métier sont réalisés en premier.
Puis les diagrammes des vues système informatique et applicative en parallèle. Des itérations successives entre ces deux vues permettent d'affiner les différents diagrammes. Par exemple, la réalisation d'une maquette permet de préciser la spécification des cas d'utilisation avec les utilisateurs. De même, l'identification des classes participantes peut permettre de compléter le modèle du domaine affiné.

L'expression et l'analyse des besoins se matérialisent par des documents Word de différents niveaux, dans lesquels on insère des diagrammes UML produits à l'aide de l'outil PowerAMC. Les plans types de ces documents sont disponibles depuis le site développement Web : http://www.dsi.cnrs.fr/bureau_qualite/developpement-web/guides-modeles/guides-modeles.asp :

- Note de cadrage (inclut des diagrammes de la vue métier)
- Exigences fonctionnelles (inclut des diagrammes de la vue système informatique)
- Conception de l'interface utilisateur (inclut des diagrammes de la vue applicative)

Dans l'outil PowerAMC, les différents diagrammes prennent place dans des packages qui structurent le modèle. Le modèle type DSI (*Modèle – application web.moo*) se présente ainsi :


1 - Vue métier
1.1 - Acteurs
1.2 - Les processus métier
1.4 - Modèle du domaine
1.5 - Modèle du domaine affiné
1.3 - Nom d'un processus métier
2 - Vue système informatique
2.1 - Contexte statique
2.3 - Les cas d'utilisation
2.2 - Contexte dynamique (collaboration)
2.2 bis - Contexte dynamique (séquence)
3 - Vue applicative
3.1 - Navigation
3.2 - DCP 1 - Nom d'un cas d'utilisation

Vue métier

La vue métier permet de mieux connaître et comprendre les processus dans lesquels va s'intégrer le futur système informatique.

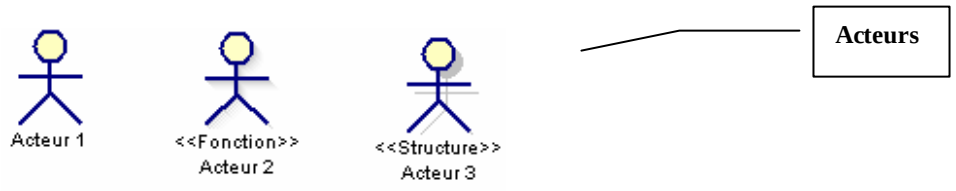
1. Les acteurs

Pour démarrer, les acteurs métier concernés par le système d'information sont identifiés.

1.1. Modélisation des acteurs métiers

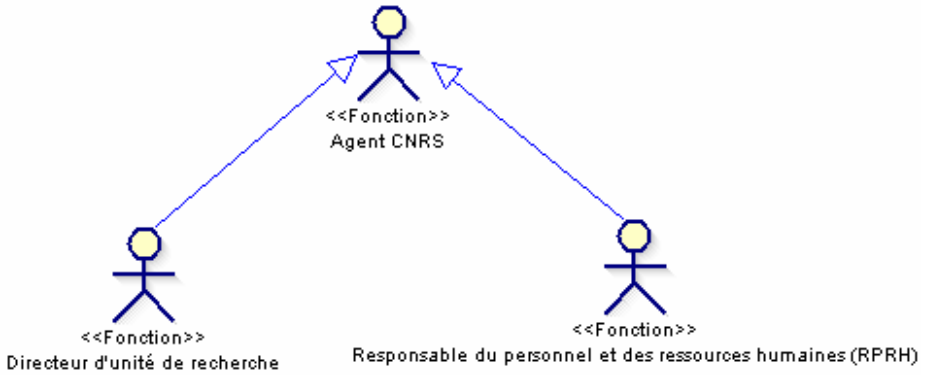
- **Identifiez les acteurs**
  - Les **acteurs** sont décrits par une abstraction ne retenant que le rôle qu'ils jouent dans le cadre du métier étudié.
  - Il est possible de préciser si le rôle est attribué à une fonction au sein du CNRS ou une structure.
  - Le nom de l'acteur commence par une majuscule. Entre parenthèses, mettre à la fin du nom de l'acteur une abréviation si possible connue au CNRS.

Exemple :

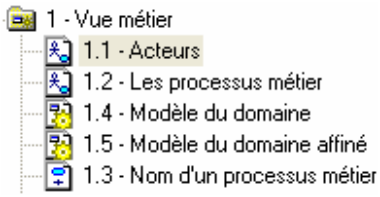


- **Identifiez les relations de généralisation entre acteurs**
  - Si un ensemble d'acteurs jouent le même rôle, on peut créer un **acteur généralisé**, qui permet de factoriser ce rôle commun.

Exemple :



1.2. Utilisation de PowerAMC : diagramme de cas d'utilisation


1 - Vue métier
1.1 - Acteurs
1.2 - Les processus métier
1.4 - Modèle du domaine
1.5 - Modèle du domaine affiné
1.3 - Nom d'un processus métier

Dans le package *« Vue métier »*, un diagramme de cas d'utilisation nommé *« Acteurs »*, contient les acteurs du système, représentés sous forme d'acteurs PowerAMC.
Dans le cas de fonctions CNRS, stéréotypez les acteurs <<Fonction>>.
Dans le cas de structures, stéréotypez les acteurs <<Structure>>.

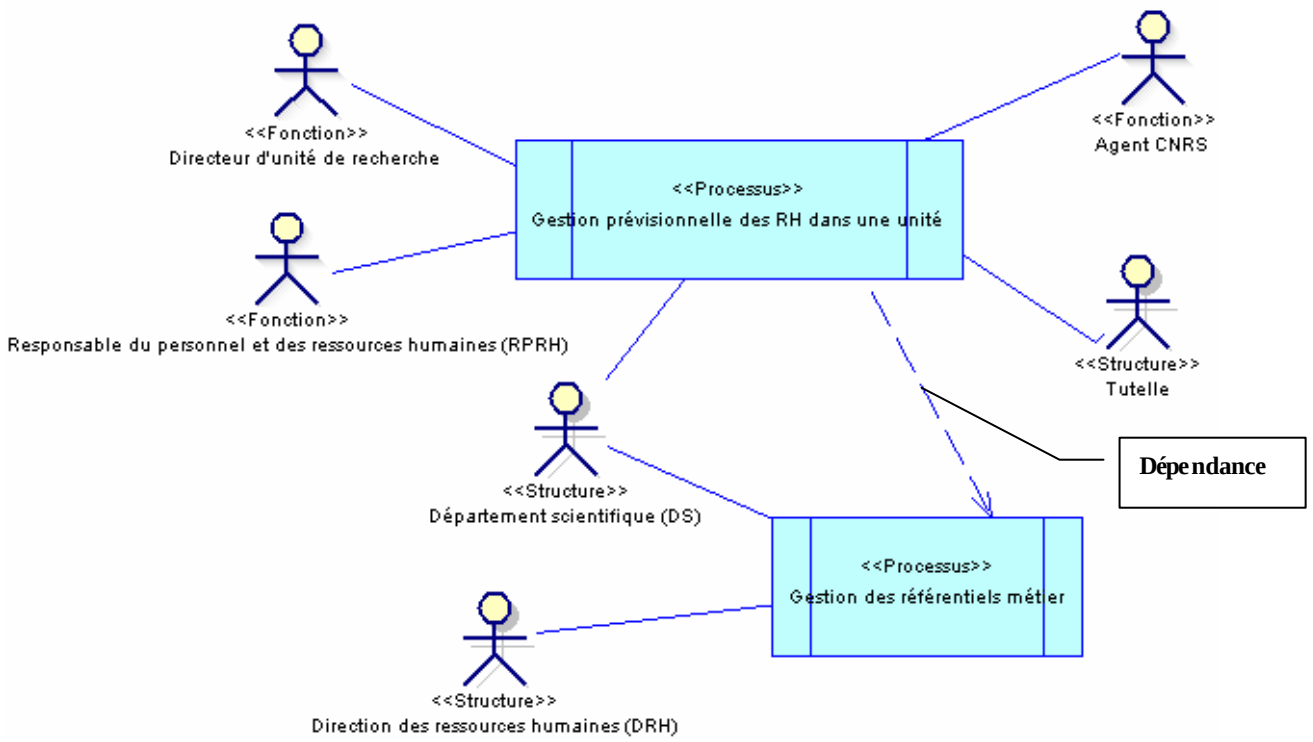
2. Les processus métier

A ce niveau, il s'agit dans un premier temps de décrire les fonctions de l'organisme qui sont au cœur de son métier, les acteurs externes concernés et les échanges entre processus.
Dans un deuxième temps, les processus les plus significatifs pour le projet sont détaillés sous forme d'événements déclencheurs et d'enchaînements d'activités.

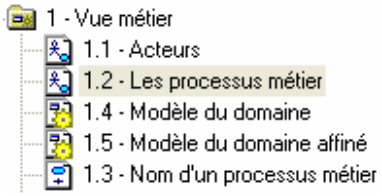
2.1. Modélisation des processus métier

- **Identifiez les processus métier du domaine étudié**
  - Le domaine étudié est décrit sous forme de processus métier et d'échanges avec les acteurs externes. Des dépendances entre processus peuvent être représentées.

Exemple :



2.2. Utilisation de PowerAMC : diagrammes de cas d'utilisation


1 - Vue métier
1.1 - Acteurs
1.2 - Les processus métier
1.4 - Modèle du domaine
1.5 - Modèle du domaine affiné
1.3 - Nom d'un processus métier

Dans le package *«Vue métier»*, un diagramme de cas d'utilisation nommé *«Les processus métier»*, contient les processus, représentés sous forme de cas d'utilisation PowerAMC, stéréotypés <<Processus>>.

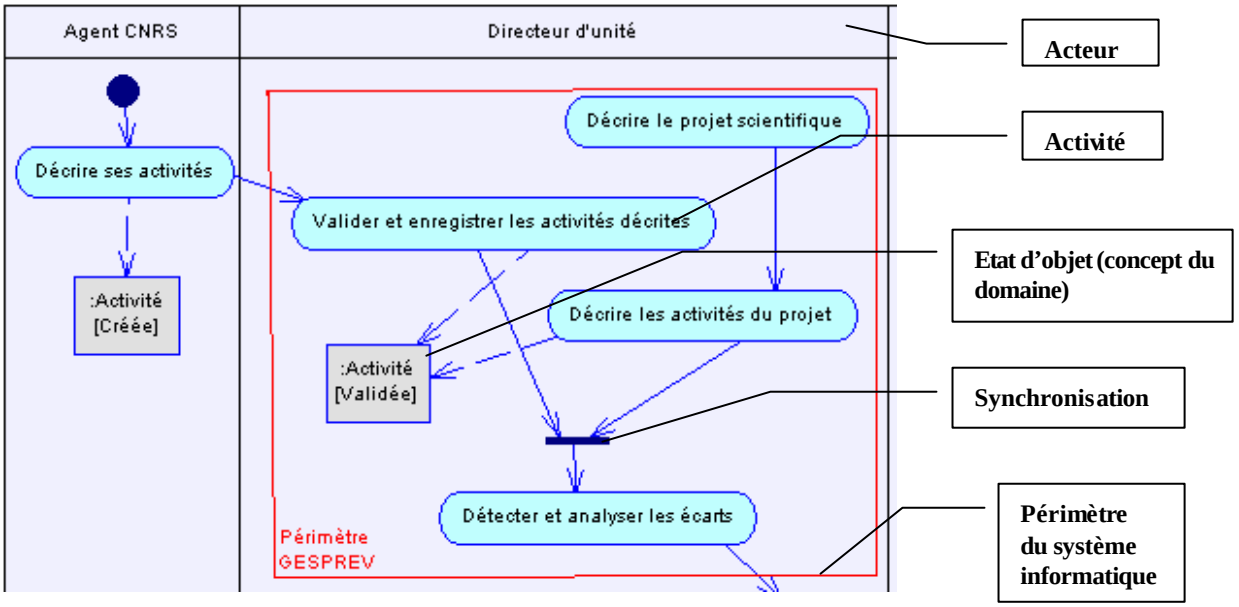
Les acteurs sont copiés sous forme de raccourcis depuis le diagramme des acteurs de la vue métier.

2.3. Modélisation des activités concourant au processus métier étudié

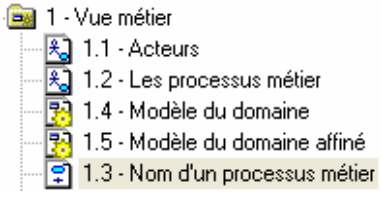
- **Identifiez les activités**

  - Le processus est décrit sous forme d'un enchaînement d'**activités**.
  - Des **transitions** relient les activités. Des **synchronisations** représentent l'attente de plusieurs événements pour la réalisation d'une activité. Des **décisions** représentent des choix de réalisation d'une ou l'autre activité en fonction de **conditions**.
  - Chaque diagramme commence par un **début** (un seul début) et se termine par une ou plusieurs **fin(s)**.
  - Des **états d'objets** peuvent être représentés afin de modéliser les concepts sous-jacents (ces concepts permettent d'initialiser le modèle du domaine, cf. chapitre suivant).
  - Si le processus est organisé, on peut modéliser les acteurs participants au processus ainsi que l'enchaînement des opérations de cette **procédure**.

Exemple :



2.4. Utilisation de PowerAMC : diagrammes d'activités


1 - Vue métier
1.1 - Acteurs
1.2 - Les processus métier
1.4 - Modèle du domaine
1.5 - Modèle du domaine affiné
1.3 - Nom d'un processus métier

Dans le package «*Vue métier*», un diagramme d'activités est créé par processus que l'on souhaite décrire en détail.

Le nom de chaque diagramme reprend le nom du processus qui est décrit. Le diagramme contient l'enchaînement des activités du processus.

Pour représenter les acteurs dans les colonnes, créez des unités d'organisation (reprendre si possible l'abréviation du nom d'un rôle, d'une fonction ou structure : pour des raisons de taille de la colonne).

Pour créer un état d'objet (concept métier), créez une nouvelle classe dans le diagramme de classes «*Modèle du domaine*». Copiez cette classe et collez la en raccourci dans le diagramme d'activités : un état d'objet se crée sur le diagramme. Saisissez le nom de l'état dans la fenêtre « *Propriétés de l'état d'objet* ».

On peut modéliser la référence à un autre processus, en le représentant sous forme d'activité, stéréotypée <<Processus>>.

Matérialisez le périmètre du système en entourant avec un élément de dessin les activités qui vont être supportées par le système informatique.

3. Le modèle du domaine

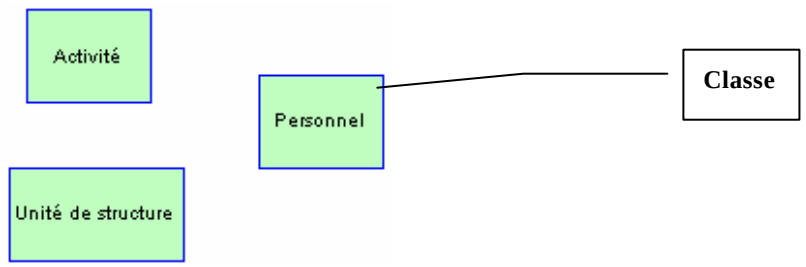
Le modèle du domaine permet de représenter les concepts du domaine, les objets métier, c'est-à-dire les informations créées, transformées ou manipulées par les experts du domaine. Les concepts sont modélisés dans un diagramme de classes.
L'objectif de ce diagramme est que l'expert du domaine y retrouve le vocabulaire de son métier. Un concept fait partie du domaine s'il est nécessaire à la compréhension du problème et s'il répond à des exigences fonctionnelles visibles par un utilisateur.

Les concepts liés à la mise en œuvre du système informatique (contraintes techniques par exemple) ne doivent pas apparaître à ce niveau, il ne s'agit pas de décrire le modèle de données de l'application mais d'identifier les principaux concepts ainsi que leur définition. On aboutit ainsi à un **glossaire du vocabulaire métier**.
Lors de la description des cas d'utilisation (chapitre suivant : vue système informatique), le modèle du domaine est affiné : complété par de nouvelles classes ou attributs, par des définitions et règles de gestion, par les types et tailles des attributs...

3.1. Modélisation des concepts du domaine

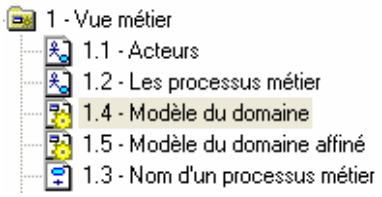
- **Identifiez les concepts du domaine**
  - Modélisez les concepts sous forme de **classes** (convention d'écriture : majuscule à la première lettre pour les noms de classe).

Exemple :



  - Précisez la définition des concepts dans un glossaire.

3.2. Utilisation de PowerAMC : diagramme de classes


1 - Vue métier
1.1 - Acteurs
1.2 - Les processus métier
1.4 - Modèle du domaine
1.5 - Modèle du domaine affiné
1.3 - Nom d'un processus métier

Dans le package *«Vue métier»*, un diagramme de classes, nommé *«Modèle du domaine»*, contient les classes du domaine.

Vue système informatique

La vue système informatique permet de s'intéresser au contexte d'utilisation du système puis aux services rendus aux différents utilisateurs (cas d'utilisation) et d'affiner le modèle du domaine.

1. Le contexte statique

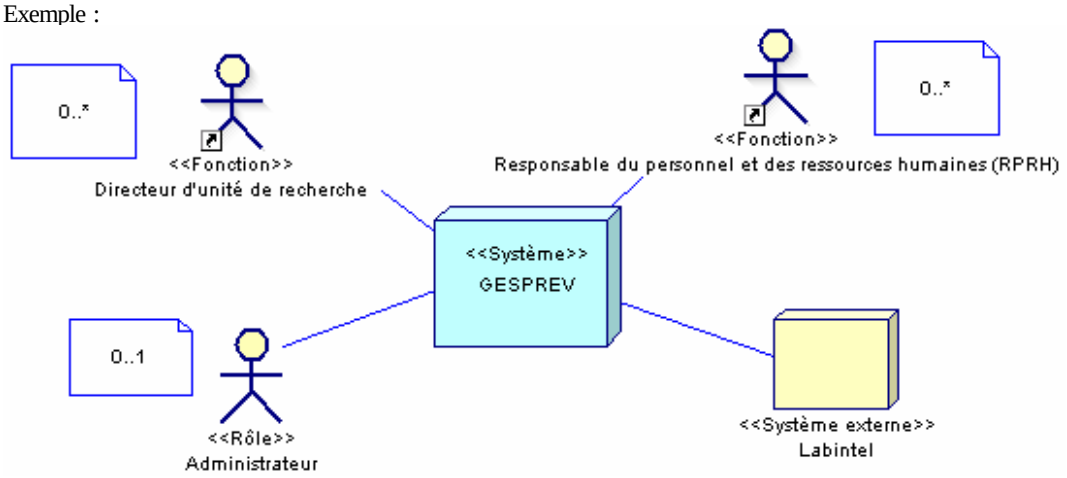
On modélise ici le **système informatique** dans son environnement, en le représentant comme une « boîte noire », en relation avec des **acteurs externes**.
Ce diagramme est utile pour avoir une vision générale externe du système. De plus il permet de donner les premiers éléments de volumétrie des utilisateurs.

1.1. Modélisation du contexte statique du système

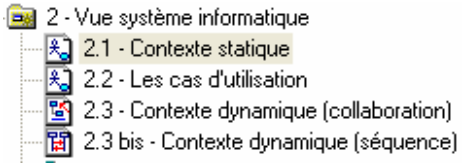
- **Représentez le système étudié**
  - Le système est représenté au centre du diagramme, avec les acteurs en relation autour.

- **Identifiez les acteurs**
  - Modélisez les acteurs externes au système : ils attendent un ou plusieurs services du système, ils interagissent avec le système par envoi ou réception de messages. Les acteurs sont décrits par une abstraction ne retenant que le rôle qu'ils jouent vis-à-vis du système.
  - Les acteurs externes peuvent être des acteurs déjà identifiés au niveau métier ou bien d'autres acteurs liés au système informatique, par exemple un *« Administrateur »*.
  - Pour chaque acteur humain, précisez le nombre susceptible d'intervenir simultanément sur le système.

Exemple :



1.2. Utilisation de PowerAMC : diagramme de cas d'utilisation


2 - Vue système informatique
2.1 - Contexte statique
2.2 - Les cas d'utilisation
2.3 - Contexte dynamique (collaboration)
2.3 bis - Contexte dynamique (séquence)

Dans le package *« Vue système informatique »*, un diagramme de cas d'utilisation nommé *« Contexte statique »*, contient le système, représenté sous forme d'un cas d'utilisation PowerAMC, stéréotypé <<Système>>. Les acteurs sont copiés sous forme de raccourcis depuis le diagramme des acteurs de la vue métier ou bien créés dans ce diagramme.

2. Les cas d'utilisation

L'identification et la description des cas d'utilisation sont certainement la partie la plus délicate de la démarche. Les conseils donnés ci-après sont issus des retours d'expérience des projets DSI et de références bibliographiques, dont en particulier : «Rédiger des cas d'utilisation efficaces» d'Alistair Cockburn.

Les **cas d'utilisation** (= use cases) permettent de capturer et décrire les besoins fonctionnels d'un système. Ils illustrent le comportement du système, suite à des stimulations externes, via la description des actions exécutées et les réponses fournies à ces stimulations.
Les cas d'utilisation permettent d'exprimer les besoins des utilisateurs, pas la solution.

Les cas d'utilisation modélisent les différents services rendus par le système aux utilisateurs. Un cas d'utilisation apporte une valeur ajoutée notable dans l'utilisation du système à l'acteur concerné. Les cas d'utilisation sont le fil conducteur du projet : au niveau de la planification, de l'analyse, de la conception puis des scénarios de tests.

Tous les cas d'utilisation sont identifiés puis chacun d'eux est étudié plus précisément. Certains cas d'utilisation peuvent ne pas être décrits dans un premier temps : lorsque l'on n'a pas suffisamment d'informations de la part des utilisateurs ou lorsque le cas d'utilisation fait partie d'une itération ou d'une version future du système.

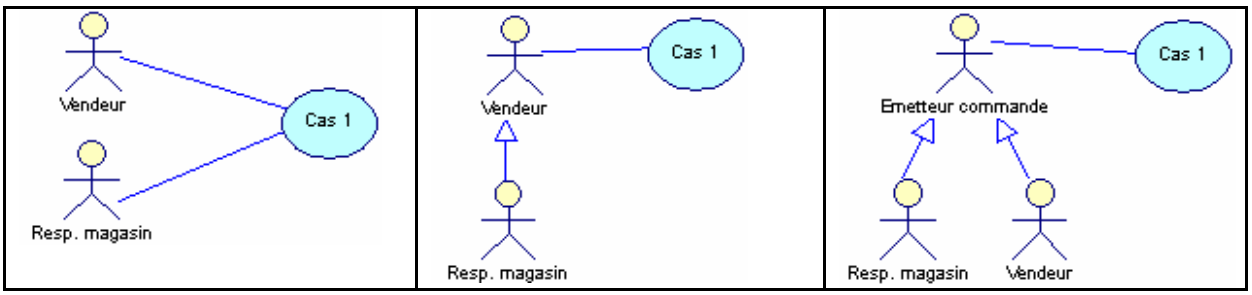
Les cas d'utilisation sont **spécifiés de manière textuelle**. Pour certains d'entre eux, la spécification peut être complétée par un diagramme dynamique simple : le diagramme de **séquence des messages**.

2.1. Modélisation et spécification des cas d'utilisation

Il s'agit ici de modéliser les fonctionnalités du système telles quelles sont perçues par les utilisateurs externes, également appelés **acteurs externes**. L'ensemble des cas d'utilisation et des acteurs intervenant dans le système est représenté sur un diagramme, appelé « diagramme des cas d'utilisation ».
L'objectif de ce diagramme est de faire clairement apparaître quels sont les acteurs qui participent aux cas d'utilisation : un acteur et un cas d'utilisation sont reliés par une **association** qui signifie « participe à ».

Sur le diagramme, on distingue les **acteurs principaux** qui réalisent le cas d'utilisation pour atteindre un objectif (récupèrent un résultat du système), des **acteurs secondaires** qui sont sollicités par le système : les acteurs principaux sont positionnés à gauche du cas d'utilisation et les acteurs secondaires à droite.

On peut utiliser les acteurs identifiés au niveau métier ou identifier de nouveaux rôles vis-à-vis du système, en utilisant la relation de généralisation entre acteurs. Les trois exemples ci-dessous sont équivalents :



- **Identifiez les cas d'utilisation**

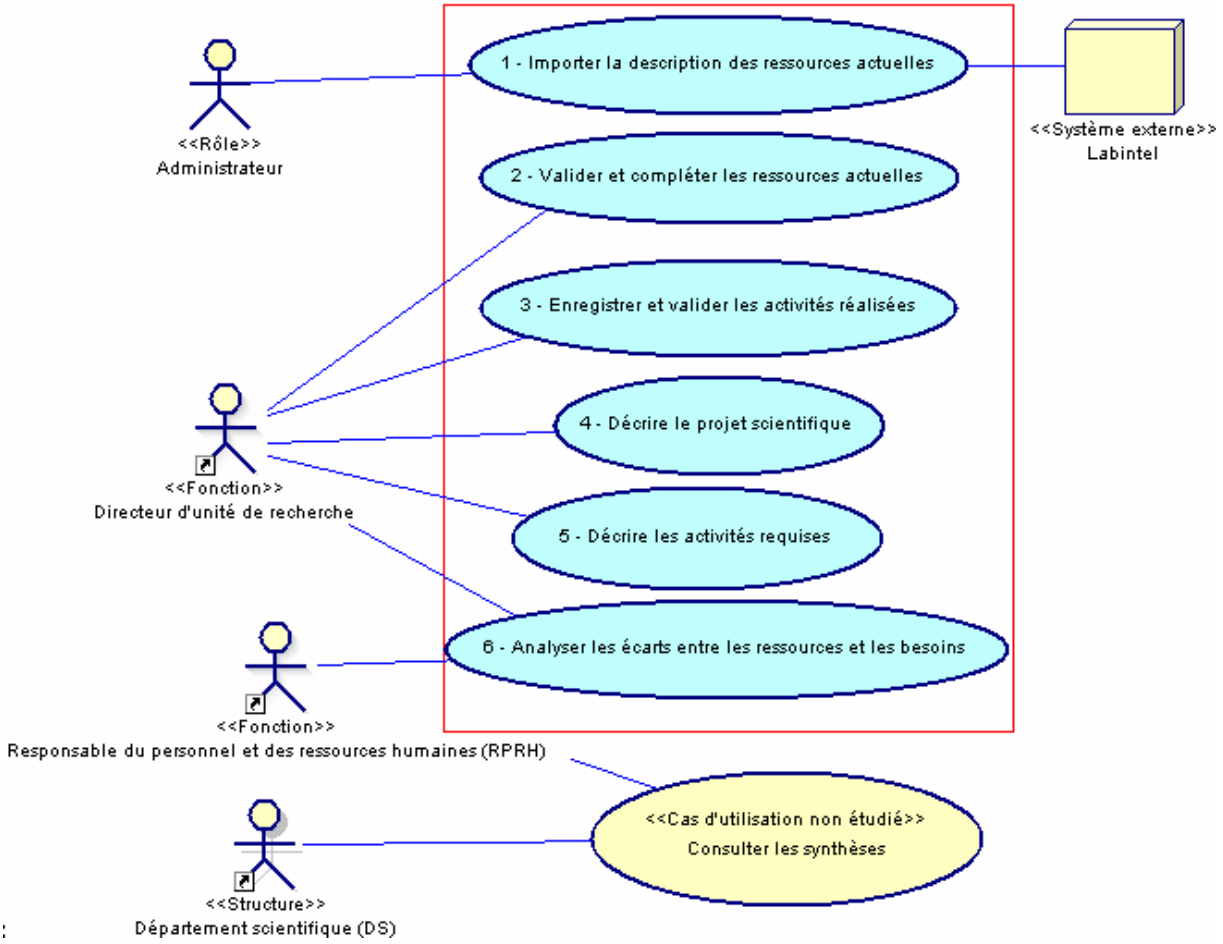
Pour chaque acteur, recherchez les différents objectifs qu'il a d'utiliser le système. Chaque cas d'utilisation est nommé par un verbe à l'infinitif, indiquant un objectif de l'acteur principal, de son point de vue.
Un cas d'utilisation correspond à « une raison d'être du système », « à une session utilisateur », « à une tâche utilisateur avant la pause café »…
Pour chaque cas d'utilisation, vérifiez qu'il fournit une valeur ajoutée notable aux acteurs principaux et contrôlez qu'un événement en déclenche l'exécution.

Uniformisez le niveau d'abstraction des cas d'utilisation :

- un cas d'utilisation doit représenter une tâche métier pour les acteurs (partir de la description des processus métier : une activité d'un processus métier donne généralement lieu à un ou plusieurs cas d'utilisation);
- un cas d'utilisation n'est pas une fonction atomique mais un ensemble de séquences d'actions (des scénarios).

Certains cas d'utilisation peuvent correspondre à des « sous fonctions informatiques » pour des raisons de lisibilité ou parce que plusieurs cas d'utilisation y font appel (voir ci-après la relation « include »)…
Mais il n'est parfois pas la peine de les rédiger : par exemple, le cas d'utilisation « s'authentifier ».

Exemple :



- **Structurez la spécification**

  Il peut être souhaitable de structurer la spécification en créant **plusieurs diagrammes de cas d'utilisation**. Les critères de regroupement des cas d'utilisation peuvent être les domaines d'expertise métier, les itérations définies, les acteurs concernés.

  Remarque : les cas d'utilisation ne s'enchaînent pas entre eux : si on veut décrire un enchaînement, le faire au niveau de la vue métier.

- **Identifiez les dépendances entre cas d'utilisation**

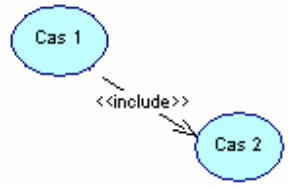
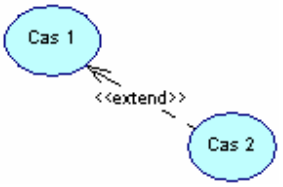
  - Relation « **include** » :

  Si un même enchaînement peut être utilisé plusieurs fois (partageable), il peut faire l'objet d'un cas d'utilisation à part, inclus dans d'autres cas d'utilisation. : ce sous cas d'utilisation permet de factoriser la partie commune de la description de plusieurs cas d'utilisation.
  Les cas d'utilisation qui l'incluent, précisent explicitement l'endroit où il est inclus dans les scénarios (cf. paragraphe suivant).
  Un cas d'utilisation inclus n'est jamais exécuté seul, mais seulement en tant que partie d'un cas plus vaste, il n'a pas d'acteur déclencheur.

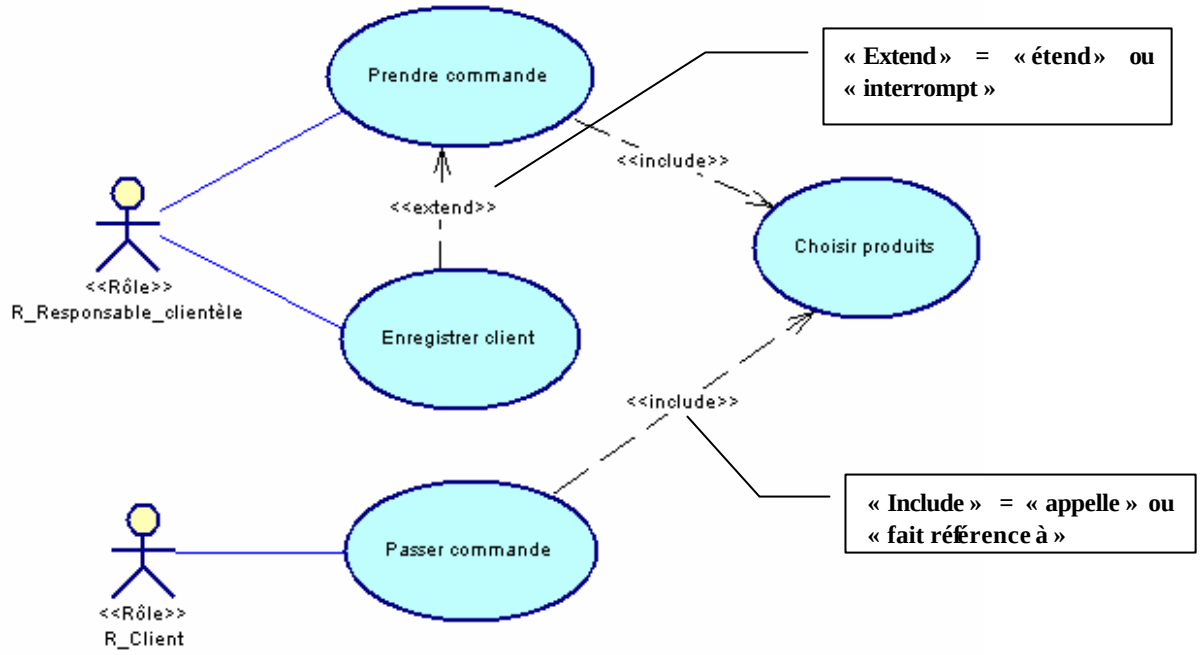
  - Relation « **extend** » :

  Un enchaînement complexe, optionnel par rapport à un enchaînement obligatoire d'actions (variante de comportement), peut faire l'objet d'un cas d'utilisation à part, qui étend d'autres cas d'utilisation.
  Les cas d'utilisation qui sont étendus par ce cas d'utilisation, précisent explicitement les points d'extension (cf. paragraphe suivant).
  Les cas d'utilisation qui sont étendus peuvent fonctionner seuls, ils ont un acteur déclencheur.
  N'utiliser cette relation qu'en cas de nécessité car elle est plus difficile à comprendre. Les situations qui la nécessitent sont :
  - lorsque l'utilisateur dispose d'un grand nombre de services asynchrones ou de services provoquant une interruption, qui ne doivent pas déranger le cas de d'utilisation de base,
  - pour compléter un cas d'utilisation dans une version ultérieure.

  Pour savoir si on utilise une relation « include » ou « extend », se poser la question :

Si le cas 1 sait quand, où, pourquoi le cas 2 doit être déclenché, alors le cas 1 « inclut » le cas 2.	Si le cas 2 sait quand, où, pourquoi il doit être déclenché, alors le cas 2 « étend » le cas 1.
	

Exemple :



Dans le cas d'utilisation « Prendre commande », on trouve :
« Le responsable clientèle choisit des produits (cas d'utilisation « Choisir produits »). »

Dans le cas d'utilisation « Enregistrer client », on trouve :
« A chaque fois que le client est nouveau quand on passe une commande (cas d'utilisation « Prendre commande »), le responsable clientèle enregistre le client. »

- **Décrivez de manière textuelle chaque cas d'utilisation**
  - La spécification doit tenir en une à trois pages de texte (document Word). Elle doit être lisible, la perfection n'est pas indispensable.
  - Un cas d'utilisation contient l'ensemble des scénarios possibles pour la réalisation d'un objectif. Un scénario correspond à l'exécution d'un ou plusieurs enchaînements, joignant le début du cas d'utilisation à une fin normale ou pas.
  - Le cas d'utilisation débute avec un événement déclencheur (première action) et se poursuit jusqu'à ce que l'objectif soit atteint ou abandonné et que le système assume ses responsabilités par rapport à l'interaction.
  - Lors de la description des cas d'utilisation, vous pouvez compléter si besoin est, les concepts du modèle du domaine.

La trame de la spécification d'un cas d'utilisation est la suivante :

Objectif

Présenter succinctement ce que permet de faire le cas d'utilisation.

Acteurs principaux

Donner la liste des acteurs qui déclenchent le cas d'utilisation dans le but d'atteindre son objectif et qui récupèrent un résultat du système (positionnés à gauche du cas d'utilisation dans le diagramme). Préciser leurs intérêts vis-à-vis de la réalisation du cas d'utilisation.

Acteurs secondaires

Donner la liste des acteurs qui sont sollicités par le système (positionnés à droite du cas d'utilisation dans le diagramme). Préciser leurs intérêts vis-à-vis de la réalisation du cas d'utilisation.

Préconditions

Décrire les contraintes qui doivent être vérifiées lorsque le cas d'utilisation est déclenché. En général une précondition indique qu'un autre cas d'utilisation s'est déroulé auparavant.

Postconditions

Décrire les contraintes qui doivent être vérifiées lorsque le cas d'utilisation est terminé.

Scénario nominal

Décrire la séquence normale d'actions associée au cas d'utilisation. En général, utiliser 3 à 9 actions. Numéroter les actions de manière chronologique.

1. Action 1
2. Action 2
3. …

La première action est généralement le déclencheur du cas d'utilisation. **A la fin du scénario nominal, l'objectif du cas d'utilisation doit être atteint.**

Chaque action donne lieu à une phrase. Commencer systématiquement chaque phrase par « l'acteur xxx… » ou « le système… ».

La phrase indique l'intention de l'acteur (pas ses gestes) : les détails de l'IHM ne doivent pas apparaître.

Eviter :

1. Le système demande le nom.
2. L'utilisateur saisit son nom.
3. Le système invite l'utilisateur à saisir son adresse.
4. L'utilisateur saisit son adresse.
5. L'utilisateur clique sur « ok ».
6. Le système présente le profil de l'utilisateur.

Préférer :

1. L'utilisateur saisit son nom et son adresse.
2. Le système présente le profil de l'utilisateur.

Il faut insister sur les événements entre les acteurs et le système sans décomposer le traitement effectué à l'intérieur du système (vu comme une boîte noire). Chaque action n'a de justification que si elle décrit une action protégeant ou accroissant les intérêts de l'intervenant. On peut associer une action à une partie d'une transaction :

1. l'acteur envoie la requête et les données au système.
2. le système valide la requête et les données.
3. le système change d'état interne.
4. le système répond à l'acteur en lui présentant le résultat.

…ou associer les différentes parties en une ou plusieurs actions, selon le degré de complexité de chaque partie et les lieux de rupture naturels dans le traitement. Par exemple :

Version 1 :

1. Le client saisit son numéro de commande.
2. Le système détecte que ce numéro correspond au numéro gagnant du mois, inscrit l'utilisateur et le numéro de commande comme gagnant du mois, envoie un mèl au responsable des ventes, félicite le client et lui donne les instructions nécessaires pour retirer son prix.

Version 2 :

1. Le client saisit son numéro de commande.
2. Le système détecte que ce numéro correspond au numéro gagnant du mois.

3. Le système inscrit l'utilisateur et le numéro de commande comme gagnant du mois, envoie un mèl au responsable des ventes, félicite le client et lui donne les instructions nécessaires pour retirer son prix.

Ecrire des phrases simples, montrant clairement « qui a le ballon » et montrant le processus en train d'avancer.

Mentionner le déroulement temporel uniquement lorsque c'est nécessaire. Exemple :

« A tout moment entre les actions 3 et 5, l'utilisateur… »

ou « Dès que l'utilisateur…, le système… ».

Il est possible de souligner le fait que des actions peuvent se répéter ou s'effectuer dans n'importe quel ordre. Exemples :

« Le client répète les actions 3 à 5 jusqu'à ce qu'il indique qu'il a terminé » (à placer après l'action 5)

ou « Les actions 3à 5 peuvent se produire dans n'importe quel ordre » (à placer avant l'action 3).

Il ne doit pas y avoir de « si… sinon ». Le système ne « vérifie » pas mais « valide » (en cas d'erreur il y aura un scénario alternatif, cf ci-après).

Exemple :

Eviter :

2. Le système vérifie si le mot de passe est correct.
3. Si c'est le cas, le système présente les actions disponibles à l'utilisateur.

Préférer :

2. Le système valide que le mot de passe est correct.
3. Le système présente les actions disponibles à l'utilisateur.

Faire référence explicite aux cas d'utilisation inclus ou qui étendent le cas d'utilisation courant (on peut souligner ces cas d'utilisation).

Scénarios alternatifs

Il s'agit d'identifier toutes les autres situations possibles de succès ou d'échec.

Pour chaque action du scénario nominal, se poser la question si quelque chose d'autre peut se passer : si c'est le cas, décrire les actions qui prolongent la séquence d'actions normale. Ces scénarios alternatifs ouvrent le cas d'utilisation à une possibilité de prolongement (extension).

Le titre de chaque scénario alternatif est la condition nécessaire pour que se déroule ce scénario.

Chaque scénario alternatif est numéroté en faisant référence aux numéros du scénario nominal précédemment décrit. Ajouter une sous-numérotation 1, 2, 3… (et si besoin a, b, c... de nouveau ainsi qu'une indentation).

Un scénario alternatif s'achève par la satisfaction ou l'abandon de son objectif. Terminer chaque scénario alternatif par « le cas d'utilisation se termine » ou « le cas d'utilisation reprend au point N » ou « le cas d'utilisation continue au point N ».

Faire référence explicite aux cas d'utilisation inclus ou qui étendent le cas d'utilisation courant (on peut souligner ces cas d'utilisation).

Exemple :

1-a : description du scénario alternatif **a** de l'action **1** du scénario nominal

1. Action 1
2. Action 2

Le cas d'utilisation se termine (échec).

1-b : description du scénario alternatif **b** de l'action **1** du scénario nominal

1. Action 1

Le cas d'utilisation continue au point 4.

2-a : description du scénario alternatif **a** de l'action **2** du scénario nominal

1. Action 1
2. L'utilisateur exécute le cas d'utilisation xxx

3. Action 3
4. …

Variantes de technologies et de données (optionnel)
Les scénarios alternatifs indiquent que ce que fait le système diffère (le quoi). Si on veut indiquer qu'il y a plusieurs façons de procéder (le comment), les lister dans ce paragraphe.
Exemple :

Scénario nominal :

7. Rembourser le client du montant des marchandises rendues.

Variantes de technologies et de données :

7a. Rembourser par chèque, virement électronique ou crédit sur les prochains achats.

Règles de gestion (optionnel)

Décrire les règles de gestion du cas d'utilisation. Les numéroter.

Exigences supplémentaires (optionnel)

Décrire les exigences non fonctionnelles du cas d'utilisation : fréquence d'utilisation du cas d'utilisation, contraintes de performance… Les numéroter.

Questions en suspend (optionnel)

Les questions à poser aux utilisateurs peuvent être regroupées dans un paragraphe spécifique en fin de description du cas d'utilisation. Les numéroter.

Exemple de spécification du cas d'utilisation « Valider et compléter les ressources actuelles » :

Objectif
Valider et compléter les ressources actuelles.

Acteurs principaux
-directeur d'unité

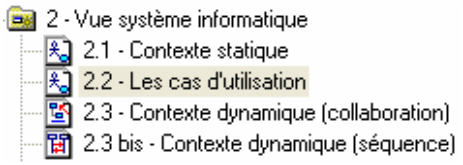
Préconditions
Import de la description des ressources actuelles effectué avec succès

Scénario nominal
1 : le directeur d'unité demande à décrire un personnel
2 : le système présente la liste des personnels
3 : le directeur sélectionne 1 personnel
4 : le système présente les informations relatives à ce personnel
5 : le directeur modifie des informations
6 : le système valide et présente les informations à jour
Le directeur répète les actions 1 à 6 jusqu'à ce qu'il indique qu'il a terminé.

Scénarios alternatifs
1a : le directeur demande la modification de la description de l'unité
  1a1 : le système présente les informations relatives à son unité
  1a2 : le directeur saisit les informations obligatoires
  Le cas d'utilisation continue au point 6
3a : le directeur demande la création d'un nouveau personnel
  3a1 : le système affiche un formulaire vierge

3a2 : le directeur saisit les informations obligatoires
Le cas d'utilisation continue au point 6
5a : le directeur demande la suppression du personnel
5a1 : le système valide qu'il n'y a pas d'activité décrite pour ce personnel
Le cas d'utilisation continue au point 6

2.2. Utilisation de PowerAMC : diagramme de cas d'utilisation


2 - Vue système informatique
2.1 - Contexte statique
2.2 - Les cas d'utilisation
2.3 - Contexte dynamique (collaboration)
2.3 bis - Contexte dynamique (séquence)

Dans le package *« Vue système informatique »*, un diagramme de cas d'utilisation, nommé *« Les cas d'utilisation »*, contient les cas d'utilisation.
Les cas d'utilisation qui ne sont pas étudiés dans la version en cours peuvent être stéréotypés <<Cas d'utilisation non étudié>> ou bien il est possible de les décrire mais on peut préciser s'ils ne sont pas réalisés dans la version en cours.

Les acteurs sont copiés sous forme de raccourcis depuis le diagramme des acteurs de la vue métier ou depuis le diagramme de contexte statique. Des associations sont établies entre les acteurs et les cas d'utilisation.

Positionnez s'il y a lieu les dépendances entre cas d'utilisation. Stéréotypez ces dépendances « include » ou « extend ».

Structurez les représentations des cas d'utilisation dans un ou plusieurs diagrammes pour favoriser la clarté de lecture par les utilisateurs.

A la DSI, nous avons choisi de ne pas renseigner la description des cas d'utilisation dans PowerAMC mais plutôt dans le document Word d'exigences fonctionnelles - cf modèle sur le site :

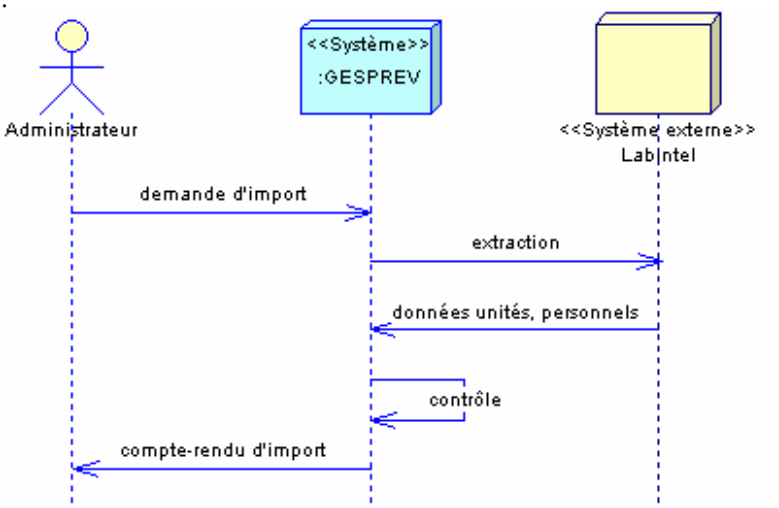
http://www.dsi.cnrs.fr/bureau_qualite/developpement-web/guides-modeles/guides-modeles.asp.

Dans l'onglet **« Classes de mise en oeuvre »** des propriétés de chaque cas d'utilisation, indiquez les classes du modèle du domaine affiné utilisées par le cas d'utilisation.
Remarque : les classes choisies comme classes de mise en œuvre dans le package « Vue métier » sont automatiquement copiées sous forme de raccourcis dans le package « Vue système informatique ». Si une classe est une classe de mise en œuvre pour plusieurs cas d'utilisation, sélectionnez le raccourci déjà créé.

2.3. Modélisation de la séquence des messages (optionnel)

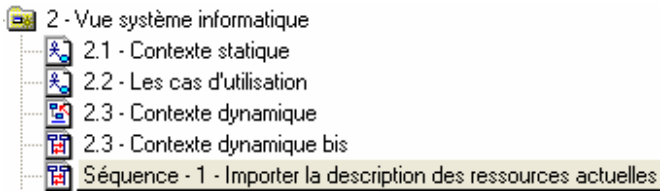
Pour certains cas d'utilisation, il peut être intéressant d'illustrer la succession temporelle des événements causés par les messages venant des acteurs. Ce diagramme est en général bien accepté par les experts métier. Il est surtout utile dans le cas où plusieurs acteurs interviennent.
Le système est considéré comme une boîte noire, l'acteur principal est à gauche, les acteurs secondaires à droite.
C'est une autre manière de représenter le scénario nominal du cas d'utilisation. Les extensions peuvent être représentées sous forme de note graphique sur le diagramme.

Exemple de diagramme de séquence du cas d'utilisation « Importer la description des ressources actuelles » :



Ne pas abuser de ces diagrammes, veiller à ce qu'ils apportent une plue value.

2.4. Utilisation de PowerAMC : diagramme de séquence


2 - Vue système informatique
2.1 - Contexte statique
2.2 - Les cas d'utilisation
2.3 - Contexte dynamique
2.3 - Contexte dynamique bis
Séquence - 1 - Importer la description des ressources actuelles

Dans le package *« Vue système informatique »*, créez un diagramme de séquence pour chaque cas d'utilisation qui le nécessite. Le nom du diagramme commence par « Séquence » et est suivi du nom du cas d'utilisation.
Représentez le système sous forme d'un objet instance d'une classe ayant pour nom le système étudié. Cet objet est stéréotypé <<Système>>.
Les acteurs sont copiés sous forme de raccourcis depuis le diagramme des acteurs de la vue métier ou depuis le diagramme de contexte statique. Les messages sont créés dans le diagramme ou copiés depuis le diagramme de contexte dynamique s'il existe.

2.5. Autres conseils pour la rédaction des cas d'utilisation

- **Les cas d'utilisation « CRUD » (= Create, Retrieve, Update, Delete) :**

Comment classer les cas d'utilisation du type Créer, sélectionner, modifier et supprimer ? Dans un seul cas d'utilisation « Gérer » ou dans plusieurs indépendants ? En principe ils sont séparés car chacun poursuit un objectif précis, éventuellement accompli par des acteurs différents ayant différents niveaux de sécurité. Mais ils surchargent l'ensemble donc il est préférable de les regrouper au sein d'un cas d'utilisation « Gérer » : le scénario nominal permet par exemple de sélectionner, modifier et enregistrer, les scénarios alternatifs permettent de créer, supprimer, imprimer. Si certains scénarios sont complexes (par exemple l'enregistrement) alors on peut en faire un sous cas d'utilisation (relation « include »).
Les droits d'accès sont décrits dans un paragraphe spécifique, en dehors de la description des cas d'utilisation.

- **Les cas d'utilisation « paramétrés » :**

Si on doit écrire des cas d'utilisation presque identiques (par exemple la recherche d'une donnée dans une liste à l'aide de critères), utiliser la même formulation (par exemple : trouver xxx à l'aide de critères yyy, triés par zzz).
Si besoin est, décrire le cas d'utilisation, par exemple, le cas d'utilisation « Trouver un quelque chose » :

1. L'utilisateur identifie les critères de recherche du quelque chose.
2. Le système trouve les «quelque chose» correspondants et affiche leurs valeurs d'affichage dans une liste.
3. L'utilisateur peut les trier à nouveau en fonction des critères de tri.
4. L'utilisateur sélectionne celui qui l'intéresse.

- **Les fins des cas d'utilisation :**

Chaque cas d'utilisation a deux fins possibles : la réussite et l'échec.
S'assurer que chaque cas d'utilisation satisfait aux intérêts de chaque intervenant.
S'assurer que l'échec de chaque cas d'utilisation appelé est traité.

- **Le niveau de précision à atteindre : travailler « en largeur » d'abord !**

Les descriptions des cas d'utilisation sont plus ou moins détaillées, selon le moment et les interlocuteurs (utilisateurs ou bien équipe de réalisation). Pensez toujours à la lisibilité pour vos interlocuteurs.
La première version présente uniquement la liste de cas d'utilisation avec pour chacun d'entre eux ses objectifs puis, par itérations, on affine et complète progressivement les descriptions.
On peut distinguer quatre étapes dans les niveaux de précision atteints :

1. Acteurs et objectifs :
   - Recenser les acteurs principaux (humain ou non humain), sur toute la durée de vie du système.
   - Dresser la liste exhaustive par acteur des objectifs pris en charge par le système.
   - Assigner des priorités : quels objectifs sont pris en charge par le système, dans quelle version.
2. Résumé des cas d'utilisation ou scénario nominal :
   - Esquisser le scénario nominal pour les cas d'utilisation à étudier. La première formulation de ce scénario peut être sous forme d'un récit.
   - Vérifier que chacun d'eux satisfait aux intérêts des acteurs.
3. Scénarios alternatifs et conditions d'échec :
   - Identifier tous les scénarios alternatifs et la liste des conditions d'échec.
4. Prise en compte des échecs :
   - Indiquer comment le système est sensé répondre à chaque type d'échec. La prise en compte des échecs peut révéler un nouvel acteur, un nouvel objectif ou de nouvelles règles métier.

- **La mise en œuvre de la démarche :**

Une première séance de travail en groupe (équipe projet + utilisateurs) peut permettre d'identifier les acteurs, objectifs, résumés des cas d'utilisation et le formalisme utilisé pour un cas d'utilisation.
Puis l'équipe projet rédige une première version des descriptions des cas d'utilisation à l'étude.
Cette rédaction est ensuite revue en groupe (équipe projet + utilisateurs).

- **Check-list pour la description des cas d'utilisation :**

Extraite et adaptée de : « Rédiger des cas d'utilisation efficaces » d'Alistair Cockburn.

Champ	**Question**
Titre du cas d'utilisation	1. L'objectif de l'acteur principal est-il formulé à l'aide d'une expression verbale active ?
Objectif	2. Le système est-il en mesure de remplir cet objectif ?
Préconditions	3. Sont-elles obligatoires et peuvent-elles être mises en place par le système ?
	4. Est-il vrai qu'elles ne sont à aucun moment vérifiées dans le cas d'utilisation ?
Scénario nominal	5. Comprend-il entre 3 et 9 actions ?
	6. Se déroule-t-il du déclencheur à la satisfaction des postconditions en cas de succès ?
	7. Autorise-t-il les bonnes variantes de séquencement ?
Chaque action de scénario	8. Est-elle formulée comme un objectif à mener à bien ?
	9. Le processus avance-t-il clairement après sa réalisation fructueuse ?
	10. Voit-on clairement quel acteur poursuit l'objectif (qui « a le ballon ») ?
	11. L'intention de l'acteur est-elle claire ?
	12. Etes-vous sûr que l'action ne décrit pas la conception de l'IHM ?
	13. Voit-on clairement quelles sont les informations transmises dans cette action ?
	14. Cette action est-elle une validation, par opposition à une vérification d'une condition ?
Scénarios alternatifs	15. Le système peut-il et doit-il les détecter ?
	16. Est-ce vraiment ce dont le système a besoin ?
Variantes de technologies et de données	17. Etes-vous sûr que ce n'est pas là simplement une alternative au scénario nominal ?
Contenu général du cas d'utilisation	18. Question aux représentants maîtrise d'ouvrage et aux utilisateurs : « est-ce vraiment ce que vous voulez ? »
	19. Question aux représentants maîtrise d'ouvrage et aux utilisateurs : « serez-vous en mesure à la livraison de dire si oui ou non, vous avez obtenu cela ? »
	20. Question à l'équipe de réalisation : « êtes-vous en mesure d'implémenter ceci ? »

3. Le contexte dynamique

Ce diagramme présente les mêmes éléments que le contexte statique mais en y ajoutant les **flux de messages** qui transitent entre le système et les acteurs externes. Ce diagramme permet de représenter globalement les interactions entre les acteurs et le système. Il est produit en parallèle à la description des cas d'utilisation.

3.1. Modélisation du contexte dynamique du système

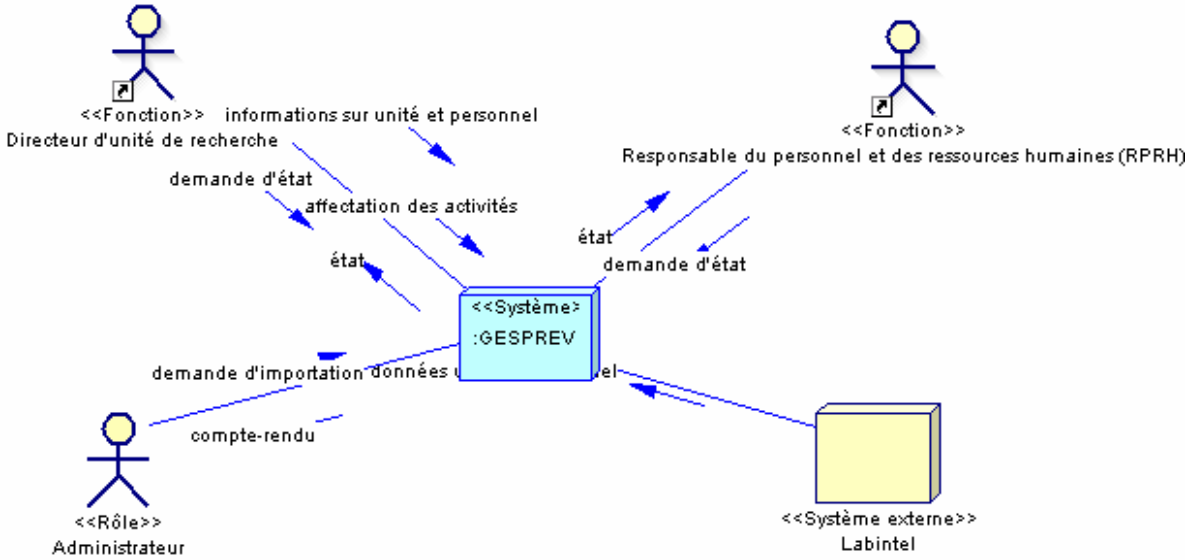
Deux diagrammes peuvent être utilisés au choix: un diagramme de collaboration et un diagramme de séquence. Les critères de choix entre l'un ou l'autre des diagrammes sont les suivants :

- Le **diagramme de collaboration** est utile lorsqu'il y a de multiples acteurs, et pour chacun, peu de flux de messages différents.

- Le **diagramme de séquence** est préférable lorsque l'on veut montrer l'enchaînement des interactions des différents acteurs avec le système.

- **Diagramme de collaboration**
  - **Représentez le système étudié et les acteurs externes**
    - De la même façon que pour le contexte statique, le système est représenté au centre du diagramme, avec les acteurs en relation autour.
  - **Identifiez les flux entre le système et les acteurs**
    - Pour chaque acteur, modélisez les messages envoyés au système et les messages que l'acteur reçoit du système.

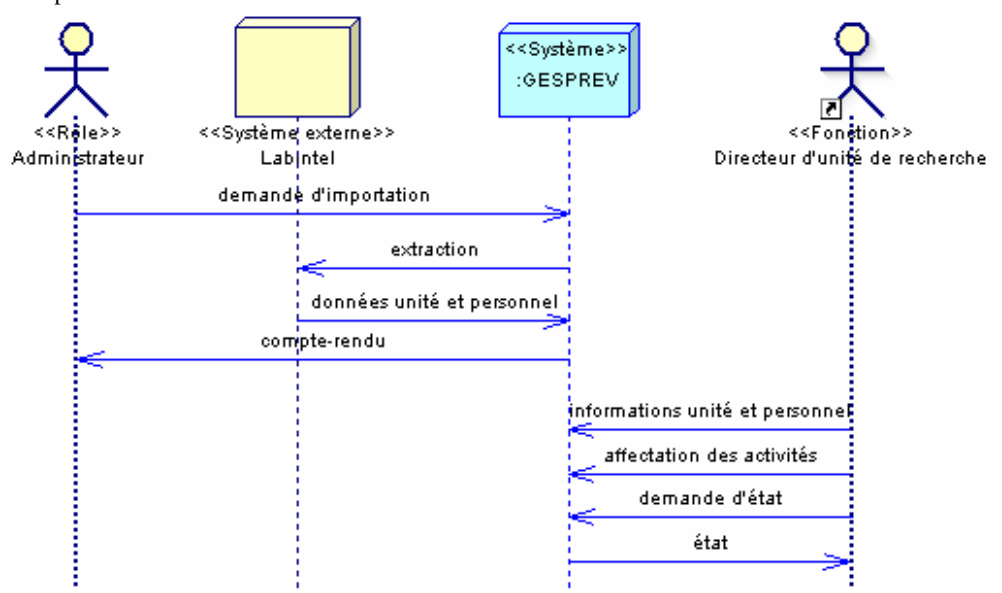
Exemple :



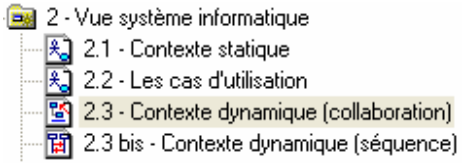
- **Diagramme de séquence**
  - **Représentez le système étudié et les acteurs externes**
    - Le système est représenté au centre du diagramme, avec les acteurs en relation de part et d'autre.
  - **Identifiez les flux entre le système et les acteurs**

- Pour chaque acteur, modélisez les messages envoyés au système et les messages que l'acteur reçoit du système.

Exemple :

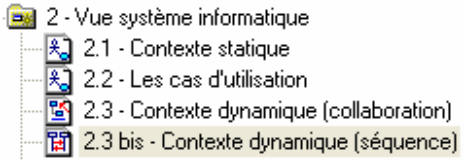


3.2. Utilisation de PowerAMC : diagramme de collaboration


2 - Vue système informatique
2.1 - Contexte statique
2.2 - Les cas d'utilisation
2.3 - Contexte dynamique (collaboration)
2.3 bis - Contexte dynamique (séquence)

Dans le package *« Vue système informatique »*, un diagramme de collaboration nommé *« Contexte dynamique »*, contient le système, représenté sous forme d'un objet, instance d'une classe ayant pour nom le système étudié. Cet objet est stéréotypé <<Système>>.
Les acteurs sont copiés sous forme de raccourcis depuis le diagramme des acteurs de la vue métier ou depuis le diagramme de contexte statique.
Créez les messages entrants et sortants du système. Dans l'onglet « Général» des propriétés de chaque message, enlevez le numéro d'ordre.

3.3. Utilisation de PowerAMC : diagramme de séquence


2 - Vue système informatique
2.1 - Contexte statique
2.2 - Les cas d'utilisation
2.3 - Contexte dynamique (collaboration)
2.3 bis - Contexte dynamique (séquence)

Dans le package *« Vue système informatique »*, créez un diagramme de séquence nommé *« Contexte dynamique »*, contient le système, représenté sous forme d'un objet instance d'une classe ayant pour nom le système étudié. Cet objet est stéréotypé <<Système>>.
Les acteurs sont copiés sous forme de raccourcis depuis le diagramme des acteurs de la vue métier ou depuis le diagramme de contexte statique. Les messages sont créés dans le diagramme.

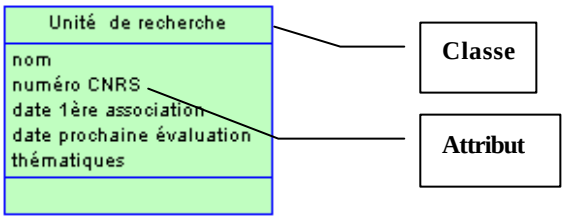
4. Le modèle du domaine affiné

A ce niveau il s'agit de préciser le modèle du domaine initialisé dans la vue métier.

4.1. Modélisation des concepts du domaine

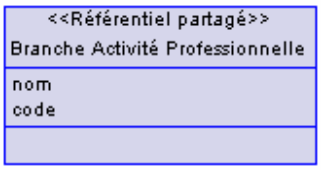
- **Identifiez les concepts du domaine**
  - Un concept possède une identité propre et des propriétés. Un concept encapsule un état et un comportement. (L'état regroupe les valeurs instantanées de toutes les propriétés d'un concept ; l'état évolue au cours du temps. Le comportement du concept décrit les actions applicables au concept ; il se représente sous la forme d'opérations.)

  - Modélisez les concepts sous forme de **classes** et les propriétés des classes sous forme d'**attributs** (convention d'écriture : majuscule à la première lettre pour les noms de classe, tout en minuscule pour les noms d'attributs et d'associations). Un attribut est une propriété nommée d'une classe qui définit les caractéristiques de cette classe.
    Exemple :



  - Parmi les classes candidates, écartez celles qui sont redondantes (indiquer les synonymes dans la documentation), trop générales ou trop spécifiques, celles qui représentent une valeur (dans ce cas c'est un attribut d'une classe) ou un comportement (dans ce cas ce serait une opération d'une classe).

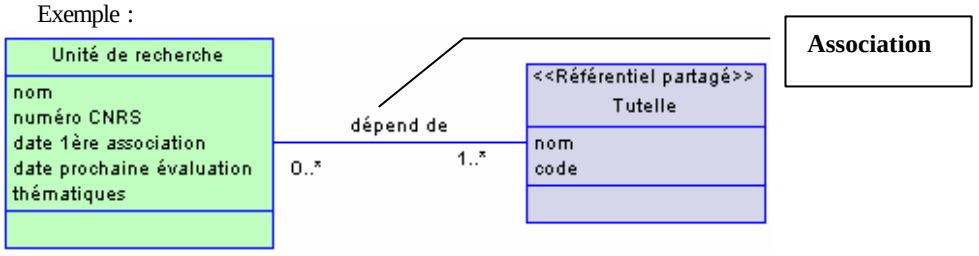
  - Parmi les classes identifiées, précisez les classes qui sont des référentiels partagés (par exemple, *BAP*, *emploi-type*…) pour le domaine étudié. Examinez la liste des référentiels partagés du CNRS pour réutiliser leur modélisation dans le projet.
    Exemple :



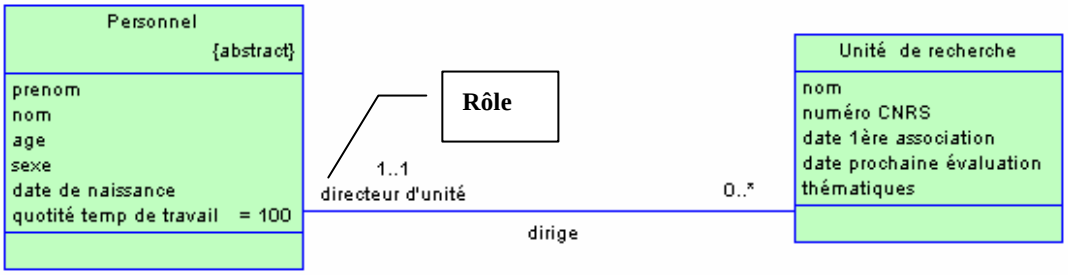
  - Déterminez les attributs qui sont dérivés, c'est-à-dire dont la valeur est déductible d'autres informations du modèle (autres attributs du même objet ou éléments externes à la classe). Explicitez la règle de dérivation des attributs dérivés dans la documentation.
    Exemple : *âge = partie entière (date courante - date de naissance)*

- **Identifiez les liens entre les concepts du domaine**
  - Modélisez les liens sous forme d'**associations**. L'association est une relation de type « est en relation avec » ou « communique avec ».
  - Nommez les associations entre classes avec un verbe conjugué précis et évitez autant que possible les verbes "à tout faire" (faire, appartenir, référencer …).

Exemple :



- Il est possible d'ajouter de l'information sur l'association pour préciser le rôle des classes dans la relation.
  Exemple :

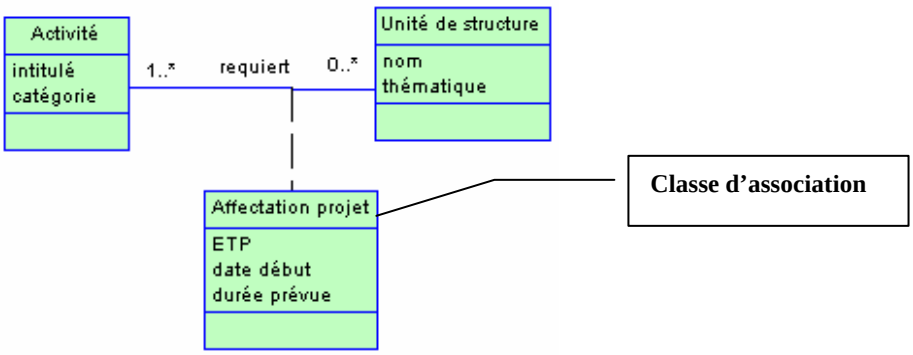


- Définissez la multiplicité de l'association (notation *min..max*) : c'est-à-dire le nombre d'instances d'une classe qui peuvent être mises en relation avec une seule instance de la classe associée.
  Exemples :

1..1	obligatoire
0..1	optionnel
0..*	quelconque
1..*	au moins 1
1..5, 10	entre 1 et 5, ou 10

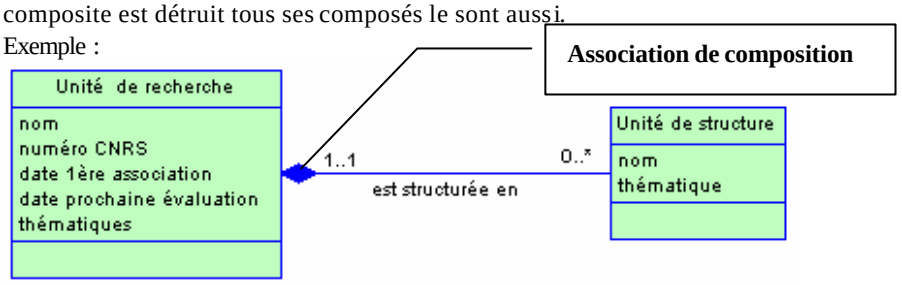
Les multiplicités sont positionnées sur l'association à l'inverse de Merise.

- Une association entre classes peut aussi porter des attributs (surtout utile quand on a une multiplicité 0..* des deux côtés). Dans ce cas l'association est aussi considérée comme une classe, appelée **classe d'association**. (Cette représentation est toutefois plutôt déconseillée : il est préférable de représenter une classe intermédiaire.)
  Exemple :



- Il existe des cas particuliers d'association : agrégations et compositions. L'**agrégation** spécifie une relation « est composé de ». La **composition** est une agrégation forte, de type « est construit avec » ou « est élaboré à base de » : un composé n'appartient qu'à un seul composite à un moment donné (la multiplicité du côté de la classe composite doit toujours être à 1..1 ou 0..1) ; si le composite est détruit tous ses composés le sont aussi.

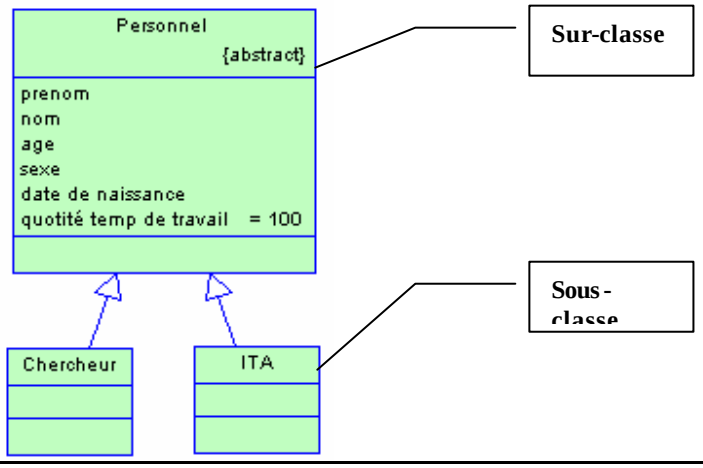
Exemple :



Ne pas abuser de l'agrégation et de la composition.

- Modélisation de l'héritage entre classes : généralisation-spécialisation. La **généralisation** consiste à regrouper des caractéristiques communes à un ensemble de classes au sein d'une sur-classe plus générale (relation de type « est un » ou « est une sorte de »). La **spécialisation** consiste à ajouter des caractéristiques spécifiques dans une sous-classe ou à adapter les caractéristiques transmises. La sur-classe peut être abstraite, c'est-à-dire qu'il n'y a pas d'instance de cette classe.

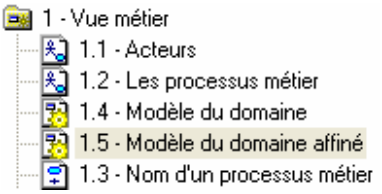
Exemple :



Ne pas abuser de la généralisation (4 niveaux maximum conseillés).

4.2. Utilisation de PowerAMC : diagramme de classes

Pour des raisons pratiques (utilisation des raccourcis des classes déjà créées dans le modèle du domaine), le diagramme du modèle de domaine affiné est créé dans la vue métier.


1 - Vue métier
1.1 - Acteurs
1.2 - Les processus métier
1.4 - Modèle du domaine
1.5 - Modèle du domaine affiné
1.3 - Nom d'un processus métier

Dans le package *«Vue métier»*, un diagramme de classes, nommé *«Modèle du domaine affiné»*, contient les classes du domaine.

- **Représentation des classes, attributs et associations**

  Dans l'onglet « Général» des propriétés de la classe :
  - si la classe est abstraite, cochez la case « Abstrait »,
  - si la classe est un référentiel, stéréotypez la classe en <<Référentiel partagé>>.

  Dans l'onglet « Notes », sous-onglet « Description » des propriétés de la classe, commentez la classe ou importez un fichier texte.

  Dans l'onglet « Attributs » des propriétés de la classe : saisissez les noms des attributs et leur type de données.
  Si un attribut est dérivé, cochez la case « Dérivé » dans l'onglet « Général » des propriétés de l'attribut.
  Documentez la règle de dérivation dans l'onglet « Notes », sous-onglet « Description » des propriétés de l'attribut ou sous forme d'une règle de gestion associée (idem : texte dans l'onglet « Notes », sous-onglet « Description » des propriétés de la règle de gestion).
  Si les valeurs d'un attribut sont connues, saisissez les dans l'onglet « Contrôles standard » des propriétés de l'attribut.

  Dans l'onglet « Général» des propriétés d'une association : saisissez le nom de l'association.
  Spécifiez les multiplicités pour les associations entre classes dans l'onglet « Détails » des propriétés de l'association. Cochez la case « navigable » des deux côtés pour supprimer les flèches sur le diagramme.

- **Structuration du modèle : création de plusieurs diagrammes de classes**

  Il peut être nécessaire de structurer le modèle du domaine : en particulier si la taille du diagramme est trop importante et en devient illisible. Des classes ayant des sémantiques proches et des relations fortes entre elles vont être regroupées.
  Chaque groupe de classes peut faire l'objet :
  - d'une page A4 dans le diagramme de classe. Si une classe doit être représentée sur deux pages, elle est copiée sous forme de synonyme graphique,
  - ou d'un diagramme de classes différent. Si une classe doit être représentée sur deux diagrammes, elle est copiée sous forme de raccourci. Les diagrammes sont nommés en fonction des groupes de classes qui y sont contenus.

  Vous pouvez aussi créer des **sous-packages**. Il est possible de transformer un diagramme en package (clic droit de la souris sur le diagramme dans l'explorateur d'objet et choisir l'option ***Convertir en package…***).

Mais attention dans PowerAMC (V 9), un diagramme ne peut pas être changé de package. Et on ne peut pas supprimer un package sans supprimer aussi tout ce qu'il contient.

Vue applicative

La vue applicative permet de donner une vision concrète du futur système grâce à une maquette, à la modélisation des principes de navigation entre les pages et d'un début d'architecture.

1. La maquette

Maquette = représentation graphique des pages, du système de navigation et donc du découpage en tâches pour l'utilisateur. Son développement ne s'effectue pas forcément dans la technologie retenue pour l'application ; la maquette est *a priori* jetable.
La maquette permet aux utilisateurs de concrétiser le résultat de leur expression de besoins. Elle est essentielle dans la phase d'expression et analyse des besoins. Elle se réalise en parallèle avec les autres modélisations, en particulier le diagramme de navigation décrit ci-après.

1.1. Réalisation de la maquette

- **Définissez l'ergonomie**
  La maquette permet de définir l'enchaînement des pages de l'application et les règles d'ergonomie. Dans un premier temps, il s'agit d'identifier les principes génériques d'ergonomie (navigation, en respectant la charte graphique CNRS) puis ensuite de définir les règles d'ergonomie spécifiques à l'application : pour cela faire intervenir l'ergonome de la DSI.
  La charte graphique CNRS doit être appliquée sur les pages de l'application. Pour cela, faire intervenir le graphiste de la DSI.

- **Développez les pages et effectuez une revue ergonomique et graphique de la maquette**
  On peut ne maquetter qu'une branche type de l'application : la plus importante ou la plus délicate.
  Il est préférable d'utiliser des données réalistes dans la maquette. L'objectif est que les utilisateurs ne se focalisent pas sur les données elles-mêmes.
  La revue ergonomique et graphique s'effectue en présence de l'ergonome et du graphiste de la DSI et de représentants de l'équipe projet : elle permet d'ajuster les règles d'ergonomie avant de montrer la maquette aux utilisateurs.

- **Organisez pour les utilisateurs une réunion de présentation et d'évaluation**
  Au cours de cette réunion, précisez leur les éléments à valider, laissez leur manipuler la maquette, recueillez leurs remarques. Pour cela :
  - soit on leur demande de réaliser une succession de tâches que l'on aura décrites dans un scénario, tâches représentatives des tâches réelles,
  - soit on leur laisse le temps de naviguer et de découvrir l'application comme bon leur semble à travers une exploration libre mais commentée de leur part.

  En plus des observations que l'on aura faites, on pourra demander ensuite aux utilisateurs ce qu'ils pensent de l'application : est-ce que la navigation leur semble intuitive, les informations faciles à trouver et s'ils se sont bien familiarisés avec l'ensemble de l'application, est-ce que l'organisation de l'application est en accord avec leur mode de travail, est-ce que les informations sont pertinentes, complètes…

- **Itérez avec la spécification**
  Vérifiez la cohérence entre la maquette et la spécification des cas d'utilisation.

1.2. Utilisation d'un outil

En interne DSI, utilisez l'outil Frontpage pour la maquette, PowerPoint ou Word. Frontpage est le plus performant au niveau graphisme. Faire appel au graphiste de la DSI.

2. La navigation (optionnel)

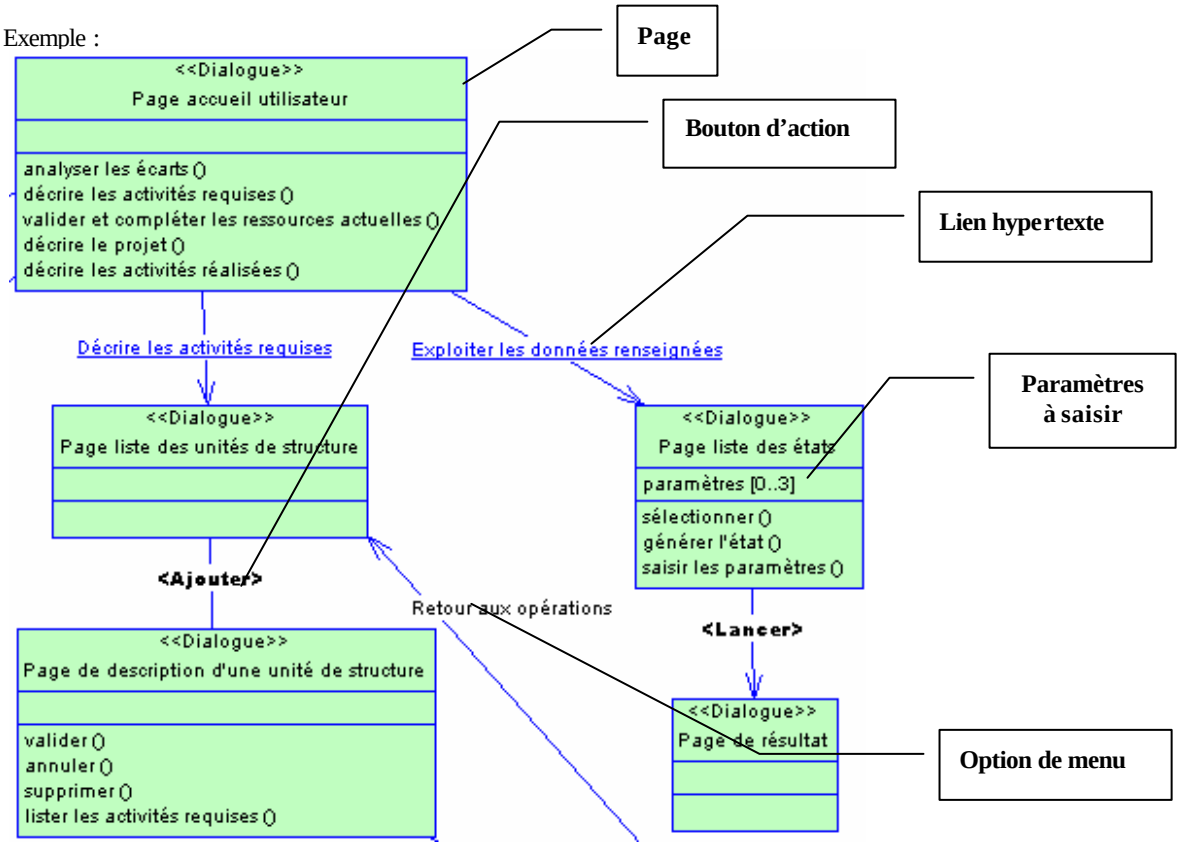
Cette modélisation s'effectue en parallèle avec la réalisation d'une maquette. Il s'agit de représenter les différents cheminements possibles des utilisateurs dans l'application. Contrairement à la maquette (qui peut ne couvrir qu'une branche de l'application), tous les cheminements possibles sont représentés dans le diagramme de navigation. Si la navigation dans l'application est très simple (peu de niveaux d'imbrication des pages), il n'est pas utile de réaliser ce diagramme.

2.1. Modélisation de la navigation

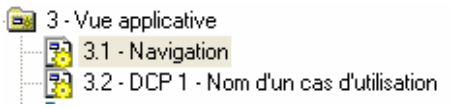
- **Identifiez les pages principales de l'application**
  - Modélisez les pages sous forme de **classes**.
  - Les **paramètres** saisis par les utilisateurs peuvent être représentés sous forme d'attributs de la classe.
  - Les **actions** proposées à l'utilisateur sur chaque page peuvent être représentées sous forme d'opérations, nommées par un verbe.

- **Identifiez les liens, boutons d'action ou options de menu**
  - Modélisez les liens hypertextes, les boutons d'action ou les options de menu qui permettent de cheminer d'une page à l'autre sous forme d'**association** entre les classes. On ne représente pas les actions qui font demeurer sur la même page.

Exemple :

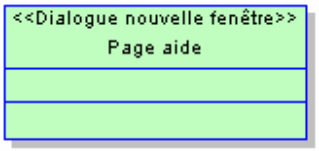


2.2. Utilisation de PowerAMC : diagramme de classes

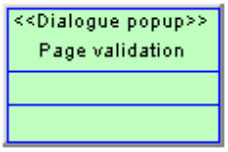


Dans le package *«Vue applicative»*, un diagramme de classes, nommé *«Navigation»*, contient les classes stéréotypées <<Dialogue>>. Il existe deux autres stéréotypes qui peuvent être utilisés :

- <<Dialogue – nouvelle fenêtre>> : pour préciser que la page s'affiche dans une autre fenêtre que la fenêtre courante du navigateur, avec la barre d'outils complète (deux fenêtres du navigateur sont ouvertes en même temps),



- <<Dialogue – popup>> : pour indiquer que la page s'affiche dans une petite fenêtre qui se superpose à la fenêtre courante, avec une barre d'outils minimale ou même inexistante (message, demande d'informations…).



Dans l'onglet « Général » des propriétés de la classe : saisissez le nom de la classe, en commençant généralement par « Page… ».

Représentez les liens hypertextes, les boutons d'action ou les options de menu sous forme d'association entre les classes.
Dans le cas d'un lien hypertexte, stéréotypez l'association <<Lien pour navigation>> et nommez la avec le texte du lien qui apparaîtra à l'utilisateur.
Dans le cas d'un bouton d'action, stéréotypez l'association <<Bouton pour navigation>> et nommez la avec le texte du bouton qui apparaîtra à l'utilisateur, entouré des caractères < et >.
Dans le cas d'une option de menu, n'utilisez aucun stéréotype.

Veillez à ce que la navigabilité entre les pages soit bonne (cochez la case « navigable » dans l'onglet « Détails » des propriétés de l'association, pour positionner la flèche sur le diagramme du côté de la page destination).

Pour ne pas afficher les multiplicités sur les associations (elles n'ont aucun sens dans ce diagramme), faites apparaître le menu contextuel du diagramme avec un clic droit de la souris sur le fond du diagramme. Choisissez l'option ***Préférences d'affichage…*** puis « Association ». Décochez « Afficher la multiplicité ».

3. Les classes participantes (optionnel - début de conception de l'architecture)

Un diagramme de **classes participantes** est réalisé pour chaque cas d'utilisation. Ces diagrammes permettent d'initialiser la conception de l'architecture de l'application sous forme de composants : il s'agit en effet d'identifier les classes « **dialogue** », « **contrôleur** » et « **entité** ».

Ces diagrammes sont construits en parallèle ou juste en suivant la description de la navigation et la réalisation de la maquette. Ils permettent de les préciser ou les compléter (ajout de classes « dialogue ») en étant exhaustif au niveau de la couverture de l'application (la maquette peut ne couvrir qu'une branche de l'application).

Le lien va également être fait avec le modèle du domaine, dans lequel on va récupérer les classes « entité », c'est-à-dire les concepts dont se sert le cas d'utilisation. Si besoin est, le modèle du domaine peut être aussi complété par de nouveaux concepts identifiés à ce stade.

3.1. Modélisation des classes participantes (« dialogue », « contrôleur », « entité ») par cas d'utilisation

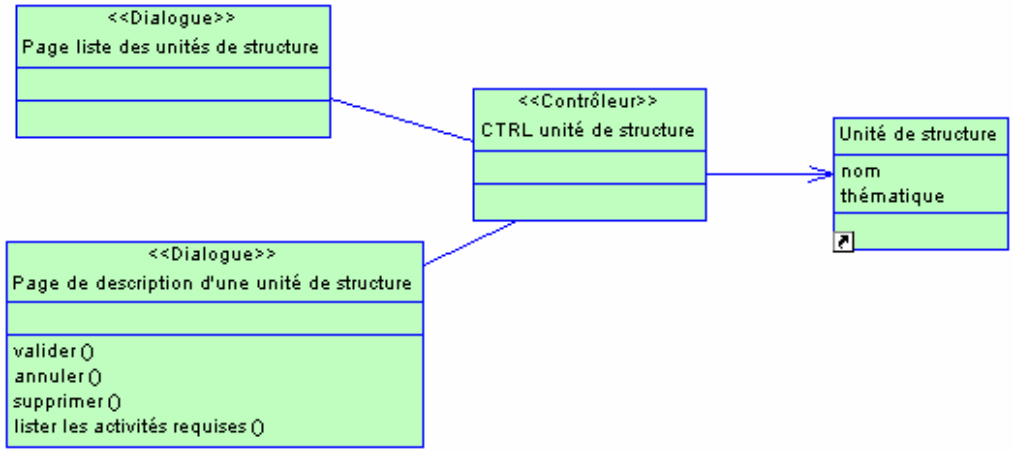
- **Identifiez les classes « dialogue », « contrôleur » et « entité »**

  Créez un diagramme par cas d'utilisation.
  - Les classes « **dialogue** » servent à modéliser les interactions entre le système et ses utilisateurs. Les classes « dialogue » sont issues du diagramme de navigation ou nouvellement créées dans le diagramme.
    Comme dans le diagramme de navigation, les paramètres saisis par les utilisateurs peuvent être représentés sous forme d'attributs de la classe. Les actions proposées à l'utilisateur sur chaque page sont représentées sous forme d'opérations, nommées par un verbe.
  - Les classes « **contrôleur** » sont utilisées pour représenter la coordination, l'enchaînement et le contrôle d'autres objets. Généralement, représentez une seule classe « contrôleur » par cas d'utilisation. Mais sur le diagramme on peut montrer qu'un contrôleur appele le contrôleur d'un autre cas d'utilisation. Il est possible de modéliser les opérations effectuées par le contrôleur, déclenchées par des actions au niveau des dialogues ou périodiquement (mise à jour de données batch).
  - Les classes « **entité** » servent à modéliser des informations durables et souvent persistantes. Les classes « entité » sont issues des concepts métier du modèle de domaine ou bien sont nouvellement créées dans le diagramme si ce sont des entités purement « applicatives », techniques (états, types d'états…).

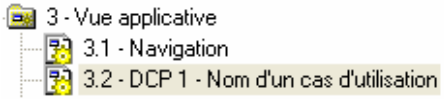
- **Identifiez les liens entre ces classes**

  Liez les classes « dialogue » à la classe « contrôleur », puis la classe « contrôleur » aux classes « entité ». Il n'y a pas de lien entre classes « dialogue », ni directement avec les classes « entité » : c'est le contrôleur qui gère la synchronisation.
  Les liens de retour à la page d'accueil ou autre retour n'ont pas à apparaître à ce niveau (ils sont représentés dans le diagramme de navigation).

  Exemple :



3.2. Utilisation de PowerAMC : diagramme de classes



Dans le package *«Vue applicative»*, créez un diagramme de classes, nommé *«DCP»* (= Diagramme Classes Participantes) suivi du nom du cas d'utilisation.
Copiez sous forme de raccourcis les classes « dialogue » déjà définies dans le diagramme de navigation ou bien créez en de nouvelles.
Créez une classe, stéréotypée <<Contrôleur>>, dont le nom commence par « CTRL », suivi généralement du nom du cas d'utilisation.
Copiez sous forme de raccourcis les classes « entité » déjà définies dans le modèle du domaine (ainsi que les associations entre elles) ou bien créez en de nouvelles.

Représentez les liens sous forme d'associations entre les classes.
Veillez à ce que la navigabilité, essentiellement entre le contrôleur et les classes « entité », soit bonne (sens du flux) : cochez la case « navigable » dans l'onglet « Détails » des propriétés de l'association, pour positionner la flèche sur le diagramme soit du côté de l'entité (signifie une écriture de données) ou du côté du contrôleur (signifie une lecture de données).
Supprimer les multiplicités sur les associations entre les classes « dialogue », « contrôleur » et « entité » (les multiplicités doivent par contre être conservées entre les classes « entité » issues du modèle du domaine). Pour cela, dans l'onglet « Détails » des propriétés de l'association, mettre à vide la rubrique « Multiplicité ».